# IMPLEMENTATION OF ENVIRONMENTAL AND SUPPLY CHAIN ANALYTICS WITH REGRESSION ON SINGLE USE PLASTICS

A PROJECT REPORT

*Submitted by*

SUPARNO BHATTA [Reg. No: RA1711003010172]

DEBANJANA CHAKRABORTY [Reg. No: RA1711003010212]

*Under Guidance of*

Dr. Poovammal E.

Professor, Department of Computer Science and Engineering

In Partial Fulfilment of the Requirements

For the Degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR 603203

MAY 2021

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR 603203

BONAFIDE CERTIFICATE

Certified that this B.Tech project report titled "**IMPLEMENTATION OF ENVIRONMENTAL AND SUPPLY CHAIN ANALYTICS WITH REGRESSION ON SINGLE USE PLASTICS**" is the bonafide work of **Mr. SUPARNO BHATTA** and **Ms. DEBANJANA CHAKRABORTY** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

| | |
|---|---|
| Dr. POOVAMMAL E. | Dr. B. AMUTHA |
| **SUPERVISOR** | **HEAD OF THE DEPARTMENT** |
| Professor | Department of Computer Science and Engineering |
| Department of Computer Science and Engineering | |

Signature of the Internal Examiner

Signature of the External Examiner

# OWN WORK DECLARATION

Department of Computer Science and Engineering

**SRM Institute of Science & Technology**

**Own Work* Declaration Form**

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

**Degree/ Course** **:** Bachelor of Technology in Computer Science and Engineering

**Student Name** **:** Suparno Bhatta

**Registration Number** **:** RA1711003010172

**Title of Work** **:** Implementation of Environmental and Supply Chain Analytics with Regression on Single Use Plastics

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

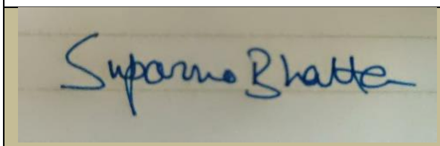
I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

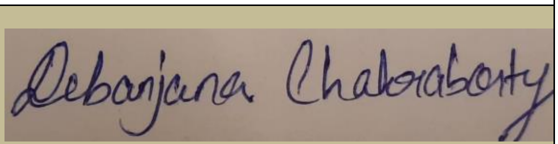
**DECLARATION:**

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.



SUPARNO BHATTA

DATE: 1 MAY 2021



DEBANJANA CHAKRABORTY

DATE: 1 MAY 2021

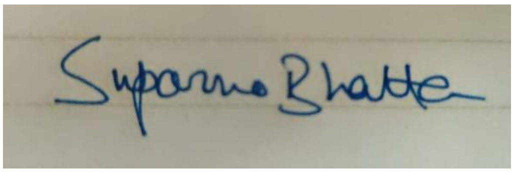
# ACKNOWLEDGEMENT

It is great pleasure for me to undertake this project. I feel highly doing the project titled – **“Implementation of Environmental and Supply Chain Analytics with Regression on Single Use Plastics”**.

I am grateful to my supervisor, **Dr. Poovammal E.**, Professor and Head of Data Mining and Big Data Analytics, Special Interest Group and my Faculty Advisor, **Mrs. D. Viji**, Assistant Professor.

This project would not have completed without their enormous help and worthy experience. Whenever I was in need, they were there behind me.

This report has been made with utmost care and deep routed interest. I hope all the readers find this thesis interesting and captivating.



SUPARNO BHATTA

# ABSTRACT

Globalization has unleashed a new era of prosperity in economic terms. This has completely changed our outlook towards lifestyle. The recent surge of consumption in the past few decades has led to the escalation in the production of goods which in turn puts a huge burden on the environment. The increasing over consumption of resources has created a sense of urgency to track the level of expenditure as well emissions from a product. One of the major concerns related to the development of green products is the cost factor that comes into play while sourcing the alternate green materials. Identifying the alternate green material increases the overall manufacturing cost for the company. The need of the hour is the need to find and develop green and sustainable products is the cost involved in replacing the raw material which produces more greenhouse gases.  There are a lot of Life Cycle Assessment (LCA) models that look into the environmental quality of a product. Similarly, there are many Supply Chain Management (SCM) models that look into the cost of the product. However very few have researched the combination of both LCA and SCM.

There is a need to establish a balance between the cost and the environmental quality of a product. This paper investigates the cost of manufacturing sustainable products by looking into existing LCA and SCM models. It is always preferred to identify alternate material in such a way that the overall cost of production should not increase much. There is definitely a need in manufacturing cheaper, cost efficient, and profitable green products for consumers and as well as manufacturers.

This research tries to analyse financial data alongside with pollution emission data of Single Use Plastics from the United Kingdom that could look into these concerns by developing an integrated model using Linear, Ridge and Huber Regression that would not only measure the environmental impact but would also keep track of the profit and supply chain demand of a product. Our project found out that Huber Regression works best with the dataset which was used. This would help manufactures to identify opportunities to reduce material and energy costs from a product system.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS AND ABREVIATIONS

| | | |
|---|---|---|
| LCA | - | Life Cycle Assessment |
| SCM | - | Supply Chain Management |
| SCA | - | Supply Chain Analytics |
| BDA | - | Big Data Analytics |
| ISO | - | International Standard Organisation |
| KNN | - | K Nearest Neighbour |
| SVR | - | Support Vector Regression |
| SVM | - | Support Vector Machine |
| RB-LCA | - | Risk Based Life Cycle Assessment |
| UK | - | United Kingdom |
| VAT | - | Value Added Tax |
| CFC 13 | - | Chloroflurocarbon-13 |
| HCFC 22 | - | Hydrochloroflurocarbon-22 |
| CO2 | - | Carbon dioxide |
| VOC | - | Volatile Organic Compounds |
| SO2 | - | Sulphur dioxide |
| PM | - | Particulate Matter |
| Kg | - | Kilogram |
| Figure. | | - Figure |
| GJ | - | Giga Joules |
| UML | - | Unified Modeling Language |
| % | - | Percentage/ Percent |

# CHAPTER

# 1. INTRODUCTION

Globalization has had a huge impact and effect on innovation, communication, technology, and lifestyle. The recent surge of consumption in the past few decades has led to the escalation in the production of goods which in turn puts a huge burden on the environment. The escalation in the commutation of people and transportation of raw materials has led to a towering consumption of fuel that has led to an increase in the pollution levels in the environment. The rise in industrialization due to globalization has led to air, water, land, radioactive, noise, light, and thermal pollution. This has resulted in an unregulated rise in prices of products and services thereby plummeting a dent in the consumption and expenditure, allowing dominance of capitalism like it never has before.

In commercial terms, a supply chain is a bureaucracy of information, people, events, organizations, and resources which necessitates in supplying a product or service to a business or a customer. Supply Chain Management (SCM) refers to the set of principles related to functions of planning, organizing, directing, and controlling the flow of goods and services which involves the transportation and warehousing of raw materials, work in process inventory, finished goods and as well as end to end order fulfilment from the point of origin to the point of consumption.

Climate change has become inevitable and the increasing overconsumption of resources combined with an imbalance in the supply chain demand due to rapid changes in the surrounding has played a huge role in undermining the situation. The raw materials which are obtained from the natural resource combined with man-made developments bloom into a complicated and useful product which may be very friendly to mankind but are responsible for the production of harmful greenhouse gases. Recent concerns amongst environmentalists have given rise to life cycle assessment that is gradually becoming an important part of the industrial process to be able to develop a safe and green product.

One of the major concerns related to the development of green products is the cost factor that comes into play while sourcing the alternate green materials that are used in place of regular ones. This research will try to establish a model that could look into the concerns by developing an integrated model that would not only measure the environmental impact but would also keep track of the profit and supply chain demand of a product. This would help companies identify opportunities to reduce material and energy costs from a product system.

### 1.1. Big Data Analytics

With growing need for resources, there is a large number of unexplored sets of raw data that are available on this planet. Most of these data are unstructured and yet to be explored. The collection of raw data to provide us with adequate information plays an important role in the application of life cycle assessment and sustainable change management. It is generally referred to as Big Data that can be defined as the field that analyses and systematically extracts information from, or otherwise deals with data sets that are too large (as much as yottabytes) or complex (such as unstructured or streaming live data). The characteristics of Big Data are 1. Volume: The amount or quantity of data generated and stored, 2. Variety: The data type and nature in the data set, 3. Velocity: The speed at which the data is being generated and processed, 4. Veracity: The data quality in the data set, 5. Value: The utility that can be extracted from the data set, 6. Variability: The rate at which the values and other different characteristics are changing. Big Data Analytics (BDA) is the use of advanced analytic techniques against enormous, heterogeneous data sets, which is an amalgamation of structured, semi-structured, and unstructured data from various sources having ferocity in volume, variety, velocity, veracity, value, and variability.

### 1.2. Life Cycle Assessment

Life Cycle Assessment [1] refers to assessment and evaluation of the lifespan of a product in terms of quality, cost, efficiency, risk, environmental and ecological factors. The leading standards of LCA include ISO 14040 and ISO 4044 that focus mainly on the process of performing an LCA. According to the international standard LCA phases are divided into a lot of steps [2].

**Goal and Scope.** This phase focuses on the definition or purpose of life cycle assessment. It focuses on identifying the functional unit that should be used to measure the environment impact. Functional unit refers to the unit of measurement and number of performances produced by products over a certain interval or period of time. Identifying the system boundary is an essential role of this phase. This helps us define compactness in a study.

**Inventory Analysis.** It is used to establish an inventor that is essential for data collection. This is a necessary factor for providing us raw details on the process or action that is going to take place. The unorganized set of data acts as a valid source of information energy, environment, economic, physical and other chemical input/output factors. It focuses on both direct and indirect data necessary for the development and maintenance of the material. It helps in analysing the quantification of a material.

**Impact Assessment.** It is used for evaluating the inventory data and understanding the respective impact the set of data can have on the material.

**Interpretation/Discussion.** It helps in identification of significant issues, limitations and results. The detailed analysis helps the stakeholders, producers, manufactures and consumers to provide thorough conclusions and recommendations that might be needed.

The Variants are mainly divided into 4 major types.

**Cradle to Grave**. This includes a complete life cycle assessment from raw phase to disposal phase.

**Cradle to Cradle.** This includes a partial product life cycle assessment from raw to production phase. The use and disposal phase are omitted in this process.

**Cradle to Cradle.** Special emphasis is placed on the disposal, recycling and reuse phase. The process starts from raw material (cradle) to disposal phase (grave) and back to reusing and raw material (cradle).

**Gate to Gate.** They majorly focus on the processing phases which can be later added to other processes.

Uses of LCA are to make Decision making tool for Climate and other environmental Impact, Social and Economic Impact and to get a Standard Accuracy for Environmental Analysis.

### 1.3. Supply Chain Analytics

There are broadly 3 types of Analytics.

**Descriptive Analytics.** It is the statistical interpretation of historical data to better identify and understand patterns, meaning, changes that occur in business.

**Predictive Analytics**. It embraces the application of big data, statistical learning approaches with predictive modelling to locate the probability of an event by manoeuvring past data.

**Prescriptive Analytics**. It is the synthesis of application of mathematical and computational sciences and advocates decision making with the help of descriptive and predictive analytics.

There is a plethora of capabilities in SCA which are mainly divided into five segments.

**Planning.** Forecasting, Compliance, Stock Keeping Unit Rationalization, Capacity verses Demand Analytics.

**Sourcing.** Supplier Risk Assessment, Commodity Research, Spend Analysis.

**Making.** Optimizing Production, Asset Utilization and Cost to Manufacture Analytics.

**Storing.** Resource Usage, Inventory Optimization and Inventory Wastage Analytics.

**Delivering.** Network Optimization, Cost-to-Serve, Logistic, Order Fill and Sales Analysis.

# CHAPTER

# 2. LITERATURE REVIEW

A Literature Survey was made to learn and understand the previous work done in the same field using different methodologies and to understand the challenges and limitations of each to not repeat them in our model.

## 2.1. Literature Review of Supply Chain Analytic Methods and Techniques

**Table 1**. SCM Techniques in Demand Forecasting

| Techniques | Remarks |
|---|---|
| **Cluster Analysis** [3] [4] | • Clustering analysis is common in the fields of pattern recognition, web development and business analytics. Algorithms like K-means, fuzzy clustering and self-organizing maps are implemented to group similar clients as regard to their conduct.<br>• This helps in better classification of different kinds or classes of data or materials. The shortfall of this technique is to recognize and pin-point the clients and groups who do not encompass a sequence or arrangement as data points. |
| **K-Nearest Neighbour (KNN)** [5] | • KNN is a category of taxonomy that brings out the similarities and likeness of the particular objects to its environment. These objects are also termed as N attributes. Each object ties or corresponds to a place in N dimensional space. |

| | |
|---|---|
| | • It identifies K objects that are nearest to the demarcated object. These aids in the creation of groups or clusters that have likely objects.<br>• KNN has been in use in demand and needs analysis of automobile spare parts in its supply chain and in the planning of Walmart supply chain. |
| **Regression Analysis** [6] [7] | • Regression Analysis generates continuous value functions which are used for prediction. It predicts the value of a dependent variable to one or more independent variables.<br>• The different types of regression are Linear, Multiple, Logistic, Weighted, Random, Polynomial, Non-parametric & Robust. Simple Moving Average model, Multiple Linear Regression.<br>• Symbolic Regression has been used with genetic programming which resulted in Symbolic Regression coming out as first.<br>• Regression and Neural Networks can be beneficial if it is used for forecasting the demand of perishable products if the sales numbers are known within the first few hours of the day. |
| **Support Vector Regression (SVR)** [8] [9] | • Support Vector Machine is the methodology which is implemented by nonlinear mapping to convert a training dataset into a higher dimension of data classes.<br>• It seeks the optimal separating hyperplane which separates the resulting class from another. Supply chains related to household and personal care have used SVMs for demand forecasting.<br>• Particle swarm optimization has been used with SVM to get a better separating hyperplane which classifies sale data and demand forecast in each cluster. It is the implementation of regression in SVMs. The logic of |

| | |
|---|---|
| | SVR is the usage of a linear regression function in a high dimensional feature. It is commonly used in Financial & Cost Prediction purposes, identification of objects, speaker and handwriting.<br>• Thus, it results in an increased accuracy. The only disadvantage is that SVRs work on problems regarding estimation and normal vs anomaly detection.<br>• SVRs have been applied with particle swarm optimization and genetic algorithms. While configuring the parameters of SVR, this approach was tested with the accuracy of Mean Absolute Percentage Error and it showed an extremely high insight by particle swarm optimization with respect to time intensity and Mean Absolute Percentage Error. |

### 2.2. Challenges of SCA Models

Companies might find leveraging cloud computing as a financial burden however data quality issues are often related to the process of data generation. Supply chain managers would find it difficult to work with data of poor quality as they rely on insights based on past data. There should be absolute completeness, consistency, accuracy and timeliness, relevancy, value, quantity, accessibility and reputation of data dimensions. In demand forecasting, most of the time is given to endogenous time complex variables and not exogenous data which has shown results of giving better insights. There is a lack of accuracy of forecasting in the supply chain due to lack of techniques available to forecast enhanced supply chain operations [10-12].

Predictive analytics is very tedious which includes different stages of development and testing. It is important to have proficient specialists of different mindsets to finish various tasks. There is a significant lack of information technology resources across companies in the supply chain network which can cause challenges in real time [13-15].

The challenges which could arise in making a multidisciplinary team can be principal agent conflicts, data sharing policies, incentive arrangements. There are major hindrances in data sharing due to outdated regulations. Security laws of data could be a major problem for multinational supply chains who are bound by laws of different countries when sharing data across the logistic chain [16-18]

It is demanding to implement real time data because the organization may react to minor changes excessively which worsens the bullwhip effect by increasing the risk in supply chain and price of warehousing. Big Data Analytics focuses on solving problems not on causation but on correlation. Indistinct profits and vagueness on the return of investment makes the investors concerned about implementing Big Data Analytics into their system. [19-21]

### 2.3. Literature Review of LCA Models

Like SCA Literature Survey, LCA Model Literature survey was also made to learn and understand the previous work done in the same field using different methodologies and to understand the challenges and limitations of each to not repeat them in our model.

**Table 2**. Techniques in Life Cycle Assessment Model

| PROPOSED MODEL | METHODOLOGY |
|---|---|
| **Process based LCA** [22, 23] | • Identifies the input and analyses the actual process of the input.<br>• Follows the ISO 14040 and ISO 4044 standard mentioned above to identify individual issues and looks into details of specific products.<br>• Here the impact stage analyses the inventory output using midpoint indicator and end point indicator. |

| | |
|---|---|
| **Economic Input/ Output** [24] | • Focuses on the economic factor of the product by taking an input and evaluating the output using matrix method.<br>• Uses linear coefficient contrast matrix method to calculate the economic transaction thus helps in calculating the supply chain and environmental emissions simultaneously. |
| **Economic Input/ Output + Process LCA** [25] | • Additional column is added to the matrix for measuring energy consumed and environmental quality of the product.<br>• Here the output from the previous industrial sector is taken as input. Each column takes in energy and emission values of a product as an input for each column. The row output from each industry is considered as an input for the other.<br>• Comprehensive method and is highly suitable for analysing the supply and demand ratio of a product. |
| **System Dynamic Model** [26] | • Focuses on the flow or rate at which the activities change by placing emphasis on the stock/ level of data that is accumulated over a significant time.<br>• It uses connectors to connect information and inputs together thus helping to regulate the flow of data. The converter helps in generating an output value.<br>• It is mostly used for finding LCA of building material needed in construction. This method is highly precise and places special emphasis on the economic and social factors of the model. |

| | |
|---|---|
| **Emission Dispersion and Exposure Assessment Model** [27, 28] | • Site-specific model that helps in quantifying the size of total damage by combining the population density with the concentration of the pollutant in a specific area. Time problem is used for analysing the health issue.<br>• The space problem arises when all the pollutant emission is aggregated irrespective of their different location specific. |
| **Environmental Matrix Model** [29] | • The row mostly consists of values of environmental load, resource, energy consumption and waste. Whereas the column consists of values obtained from the different phases namely raw material, production, manufacture and use/disposal phase.<br>• Does not give attention to the individual environment stressor i.e., the factors that affect the environment which again can bring about a change in the original value.<br>• Consist of range of value based on quality and reducing impact (0 is used for products that generate significant quantities of hazard/toxic gas during a phase. Whereas the product with the rating 4 can generate no harmful residue. |
| **Multi Objective Optimization Model** [30] | • An object specific production process model which uses linear and nonlinear program tools for LCA by optimizing the system and finding the env impact of the system.<br>• It has the capability to describe complex relationship between diff systems. The model tries to improve and |

| | |
|---|---|
| | optimize certain factors like temp pressure to test the result.<br>• It uses a nonlinear programming language to find the LCA of the product. It also optimizes a lot of physical factors. |
| **Knowledge based Approximate LCA Model**<br>[31] | • Distributed object-oriented model that identify the environment impact driver and product attribute by grouping the products into environmental characteristics and then relate it with environmental impact drivers.<br>• The modelling focuses on parts of the system where it uses different designs of a single product and tries to obtain the maximum efficiency by reducing the cost of material and increasing the environmental factor.<br>• The input of the product can vary in different ranges such as: lifetime, use time, energy source, power consumption, modes of operation. It has been designed to work on the principle that slight change in the geometry of the model can have a huge effect on the KALCAS value.<br>• This model operated on 4 modules: product information model, product LCA model, database redundancy, implementation of query facility and management support. It uses artificial intelligence using a back-propagation algorithm to minimize learning error and focus more on environmental impact. |

| | |
|---|---|
| **Neural Network** [32-34] | • Simplifies LCA calculation and reduce modelling time by calculating the LCA of a product even if complete information is presently not available.<br>• Used for evaluating the result using a case-based searching technique. Each impact factor is stored as a case based on the simple model.<br>• Here, the input is taken in the form of characteristics of the product, material, weight, feature, weight, mass and end life. The output produced by the process is the environmental impact factor. |
| **Risk Based Model** [35] | • This model is used for analysing the risk level in a model. RB-LCA is applied mostly on cost, maintenance, energy, decision making and chemical process of a product.<br>• The stakeholders are identified and the building categories are defined based on all the activities and services. Once the risks are identified, the model then classifies the data based on the severity of the risk. After which, the model identifies all the alternate options that can be used to limit the risk. RB-LCA model ensures that the cost factor is reduced.<br>• During this process, one needs to first define sustainable energy conservation, perform the process and identify the hazards. For performing risk assessment, one needs to group all the risks into categories (both qualitative and quantitative). |
| **Modular LCA** [36] | • The model is built in a C++ environment. While collecting the environmental profile of LCA, one has |

| | |
|---|---|
| | to check the consumption of the product which takes place when the product is both in stationary or dynamic phase.<br>• The operating conditions are calculated once all the processes for a single product are determined. The offline central module simulates the activity time of all single actuators that is installed on the machine for single consumption.<br>• The consumption and emission patterns are identified and the energy and mass flow are attributed based on this approach. Each matrix is used to calculate individual environmental impact. It focuses on the reusability and reconfigurability. |

### 2.4. Limitations of LCA Models

Process based LCA is one of the earliest designed methods of analysing the quality of product. While applying in the real world, it was noticed that it allows only specific product comparison. It is intensive and costly in nature [22, 23]. To remove the previous complexity an Economic Input/Output model was developed. The disadvantage to the entire model is that the data used for this process has to be considered as collective data obtained across a large area. It does not have the capability to process local/regional data separately. This was further improved by combining both the mentioned processes [24]. The Economic Input/Output + Process LCA is a combined model focused on additional environment and energy columns. However, the model once again did not focus on local data [25].

Few more models namely System Dynamic Model, Emission Dispersion and Exposure Assessment Model were developed. They were much more precise and flexible compared to previous phase models. However, all the models are dependent on the accuracy of the data. The sensitivity of the meteorological parameters is not analysed in the Emission

Dispersion and Exposure Assessment Model. This process neglects the fact that different emissions take place in different regions. The Environment matrix model faces difficulty in using the model for complicated testing and processes and can lead to several manual errors [26-29]. While using Multi objective optimization model users that lack knowledge of nonlinear programs face difficulty in using this model. It is a bit complicated and very selective in nature [30].

The Knowledge based approximate LCA model is a very sensitive model. A slight change in value can affect the entire model result. Neural Network simplifies large numbers of LCA calculations but the result of a product with new data is completely dependent on the availability of previous data. It doesn't take in consideration all data of LCA. It just focuses on specific problems. In Spite of removing a large number of traditional limits, the data used in Modular LCA depends on the accuracy of the initial data. So, any uncertainty in the initial phase can give an inaccurate result [31-36].

The information for data used in multiple processes is different for different operations of a single product. The model needs to place further emphasis on technical and logical processes. In the logical phase, new patterns of consumption and emission is noticed. In the technical process, one needs to develop a better mathematical model with increased accuracy.

# CHAPTER

# 3. INTEGRATED LCA-SCM MODEL

The study proposes a model which is an amalgamation of the Supply Chain Management Analytics and Life Cycle Assessment. According to IS0 14040 and ISO4044 Life cycle assessment (LCA) is method of assessing the quality of a product in terms of environmental and ecological factors [37]. This entire evaluation starts right from the raw material extraction phase. It journeys its way through manufacturing, processing, transportation and finally into reuse phase. Sometimes the disposal and recycling of the product is also taken in consideration to rate the sustainability of the product. A supply chain management (SCM) model necessitates data, people, events, organizations, and resources [37]. They focus on supplying a product or service to a business or a customer. They help improving the functioning, planning, organizing, and controlling the flow of goods and services. The model emphasizes on the warehousing, supplier, distributor, market, manufacturer and retailer logistics. The combined implementation of LCA and SCM is termed as Integrated LCA-SCM Model.

There are mainly 3 types of LCA-SCM models.

**Integrated Modular**. This model [38] can reveal the relative importance of supplier generated impacts with respect to the pollution impact from industry due to manufacturing. Further developments are needed in cleansing the simulation model to ease what-if analysis, the trade-offs between the operative and sustainability performance and in explication of interdependency among critical processes and improvement actions and sustainability dimensions. There is a lack of methodologies to exactly identify optimal cost with less pollution.

**Economic Input-Output Analysis.** The environmental impact of all the different input can be assessed once all the input associated with some industrial output is traced by this method

[24]. It can be used by individual companies for their products. This method has a difficulty of differentiating product types and processes used in any specific industry.

**Computer Aided Process.** This model [39] smoothly enables data exchange across supply chains but it is mostly difficult to gather data and has less software dependability. The future work includes finishing design and implementation of the XML-based web services for LCA data interchange and running experiments.

### 3.1. Purpose

Several researches have taken place in developing the perfect model that helps us understand the sustainability of the product. In the new era of economic prosperity, globalization and commercial developments have escalated the fast production of cheap goods. Most of the time these goods are available in abundance but places a huge burden on the environment.

An understanding has already been established on the existing models that focus on the environment quality of the product and also on the supply chain demand of a product [37]. Significant lack of information technology resources across companies in the supply chain network has created issues related to poor data generation and management that is essential for the development of a product [13]

The study aims to establish a globally sustainable product with less carbon foot print and more cost effectiveness. The entire test will be performed on the data on consumption of single use plastics (low density polyethylene) to check the efficiency of the model. The model will test for the effects on climate change, understanding the ecological, environmental quality of the planet and the health sector.

### 3.2. Scope

One of the major concerns related to those models were that they sounded very technical in theory but didn’t produce any optimal results when tested practically. There has been severe lack in novel methodologies that could identify the decisional areas that are necessary to differentiating product types, quality and their sustainability when used in any specific industry. They were either lagging in the technical or logical processes that is essential to identify the new patterns of consumption, emission, profit margin and also accuracy of the model. Majority of the times the data set is aggregated irrespective of them being dependent on certain factors like temperature, pressure or location specific [6-8]. Poor quality of data has been a huge issue in the past. Thus, it is essential to develop an integrated model that keeps the following attributes in mind. They include completeness, consistency, accuracy and timeliness, relevancy, value, quantity, accessibility and reputation of data dimensions. So, this project aims to solve these challenges and errors.

# CHAPTER

## 4. UML DIAGRAM

The UML Diagram was made to ease the concepts of the mode of job done by the proposed model.

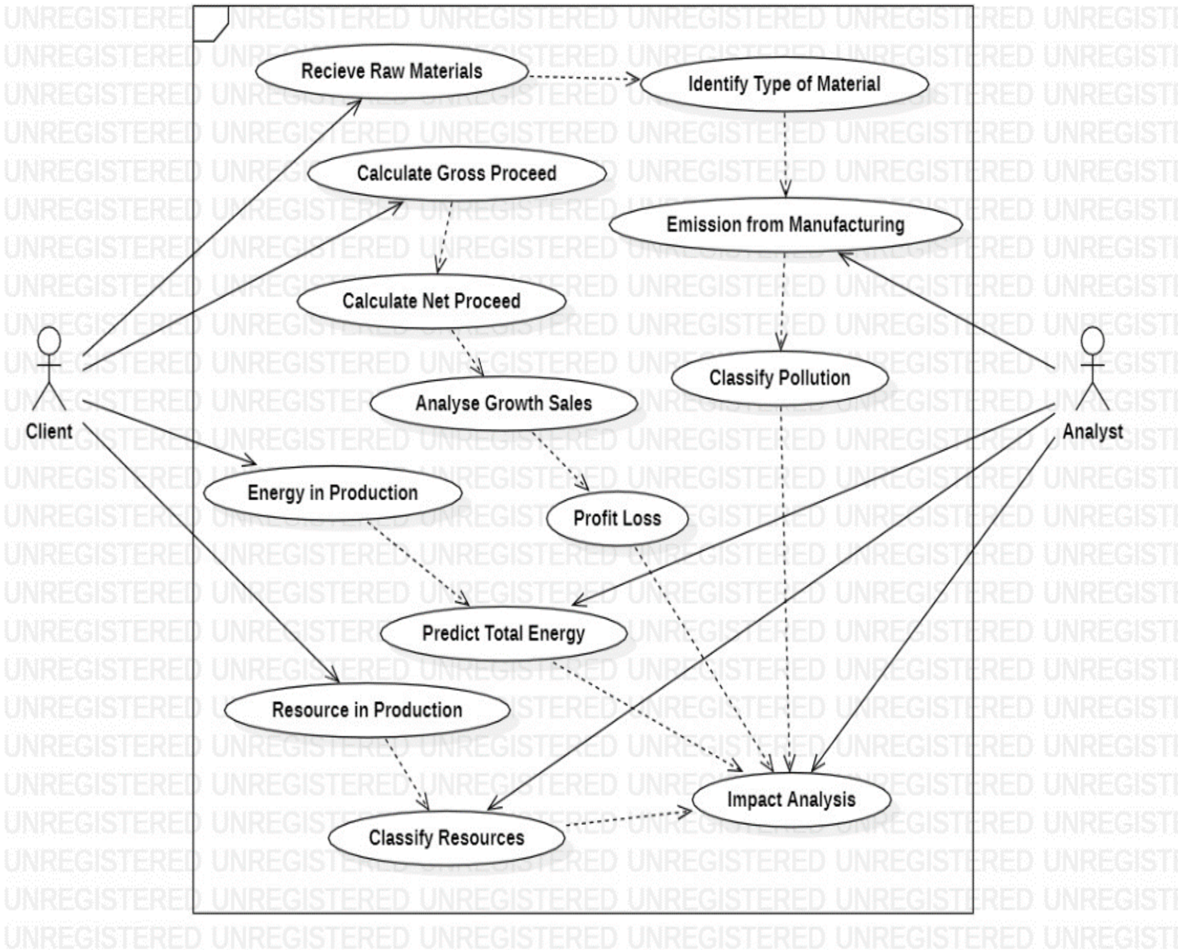



**Figure. 1.** UML Diagram for the proposed model

# CHAPTER

## 5. ARCHITECTURE DIAGRAM

The Architechture Diagram was made to ease the concepts of the modules and divide the contents of the project accordingly.

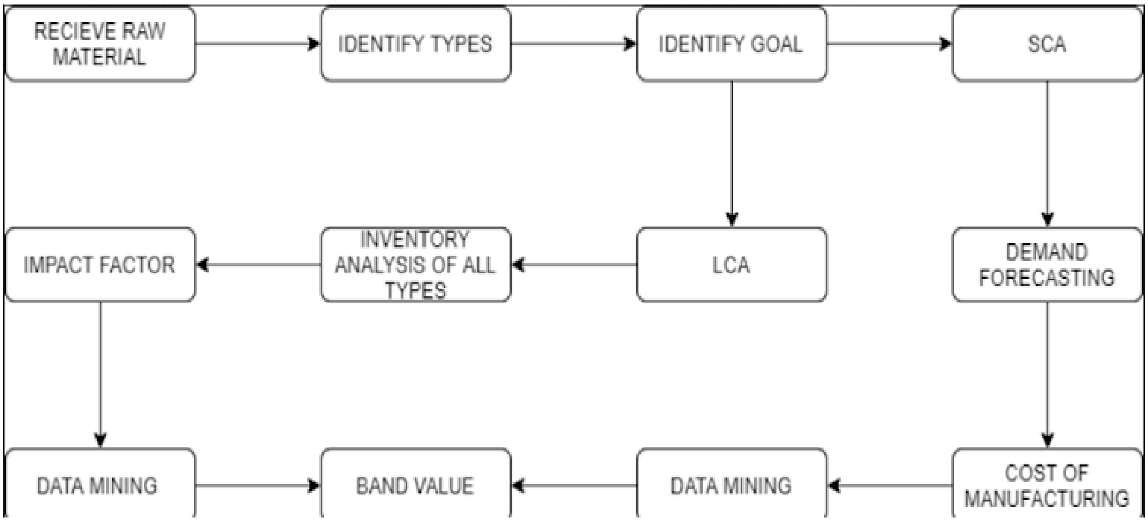



**Figure. 2.** Architecture Diagram for the proposed model

# CHAPTER

# 6. MODULE DESCRIPTION

Here is the module description of the proposed model that is being followed in this project.

**Receive Raw Material.**

The client will send an order or a request with the types of raw materials they use and what could be their future orders of raw materials and to make what kind of plastic.

**Identify Types.**

The model then tries to identify the types of raw materials and financials they have which is the input.

**Identify Goal.**

The model sets a goal as per the requirement of the client as in which type of plastic to make with how much less carbon footprint and emissions.

**SCA.**

The Model then uses Ridge, Linear and Huber Regression for Supply Chain Analytics to predict how less can it cost.

**Demand Forecasting.**

The Model tries to forecast the demand using Ridge, Linear and Huber Regression to predict the demand of the product.

**Cost of Manufacturing.**

The Model tries to figure out the cost associated with manufacturing using Ridge, Linear and Huber Regression to predict cost of manufacturing.

**Data Mining.**

Data Mining were done in Excel, Google Sheets and Python using Numpy Library.

**LCA.**

The Model then uses Ridge, Linear and Huber Regression for Life Cycle Assessment in Cradle to Grave Method to predict how less emissions can it produce.

**Inventory Analysis.**

It is a process of understanding the product stock, which is implemented by regression methods.

**Impact Factor.**

It is used for understanding inventory data, which is also implemented by regression models.

**Band Value.**

A comparison is made with SCA phase and LCA phase then the model tells us how it can give profits making better, greener plastics than before.

# CHAPTER

## 7. DATABASE COLLECTION

The Data collection for testing the LCA-SCM model is essential to understand the feasibility of the model. Very few readymade data sets are available for testing on an Integrated LCA-SCM model so a new dataset was created for this testing process. The collection procedure for the integrated model was different for both LCA and SCM phase.

The production of single use plastic (low density polyethylene) data is being used because they presently have the most demand in the global market. Their versatile usage in every aspect of our life has made them a profitable product in market. They are also the largest emitters for greenhouse gasses which are responsible for global warming. The data collection of SCM focused on the sales aspect. This was obtained from the first reporting period for the six months (October 2015-April 2016) of single use carrier submitted to the government of England. This Data [40] includes net expenditure in manufacturing the plastic, gross expenditure in selling the plastic and the value added tax by the government on single use plastic.

The SCM data was combined with self-created dataset for LCA. values were calculated using no of single use plastic and the value for one unit production or emission based on values given by a technical report on Cradle-to-Gate Life Cycle Inventory of Nine Plastic Resins & Four Polyurethane Precursors August 2011. The values of LCA focuses from Cradle to Gate phase. This data includes values of non-renewable resources used and energy consumed during the process. It also includes various green house and non-greenhouse pollutants emitted into the atmosphere during the manufacturing, process and transportation phase of single use plastics.

# CHAPTER

## 8. IMPLEMENTATION

The implementation was done with the help of the UML and Architecture diagram using a free software – Google Colab, where Python Programming language was used to carry out the project.

**LCA Phase**.

Here the environmental quality of each raw material is tested based on the amount of pollutant emission during the manufacturing process. Initially the companies under evaluation are individually tested for the energy consumption during the construction/ manufacturing phase and transportation phase. Both the phases are also responsible for greenhouse gas emission the extraction of raw materials for the product often involves lot of energy consumption and excessive emission of harmful pollutants. This data helps us calculate the impact factor on the quantity of non-renewable resources exploited during the process. Finally, the emissions from each company are classified depending on their effects on climate change, health impacts, greenhouse emission and ecological damages on emission, climate change, greenhouse gases. Once the impact factors are identified, a conclusion is drawn on the environmental quality of the product.

**SCM Phase.**

Simultaneously the data is then analysed for the cost factor of the product. It first enters the planning phase where a draft is made on the cost of manufacturing of the material and cost of selling the product to companies. The profit and loss is forecasted based on the above mentioned two factors. Sometimes the profit or loss value may differ depending on the additional costs such as operational cost, government tax or third-party cost. Thus, it is important to keep these data in mind during calculation of gross expenditure of the company. Once the sales data is tested for any external deviation the planning the total loss or profit of a company is observed.

### 8.1. Methodology

**To create a user dataset on single use plastic.**

- Obtained a data set containing a list company, no. of single use plastic consumed by the company, Net Proceed, VAT and Gross Proceed.
- Calculated the amount of energy consumed while manufacturing those single use plastics.
- Calculated the amount of natural resources used while manufacturing those single use plastic.
- Calculated the amount of emission from pollutants.
- The dataset is ready.

**Data set cleaning.**

- Look for the missing or irrelevant values
- Look for duplicate values and remove them
- Converted the data types from nominal to numerical
- Replaced missing values with 0

**Perform Data Mining using Scikit learn.**

- Opened Google Colab notebook.
- Imported the dataset in the drive and mount the Google Colab in the same drive.
- Set the path in python for accessing the dataset.
- Took all the inputs required using feature and set the target values for prediction.

- Imported Numpy library for converting the data from csv and to analyze data in table format.
- Set the training and testing data and split them based on 70:30 ratio.
- Imported scikit learn library to perform data mining in Python.
- Performed Regression analysis using Linear Huber and Ridge Regression.
- Checked for model score to obtain the accuracy of the mining technique.
- Obtained the prediction values.

**Graph Visualization.**

- Compared the real and predicted values obtained from the target result.
- Visualized the results using line graph.
- Performed impact analysis based on Ecological System Effect, Resource Depletion, Human Health, Overall Profitability.

### 8.2. Algorithms Used

**Linear Regression.**

- Helps in predicting the net profit/loss of a company based on the sale of the raw materials for manufacture purpose.
- Gives good results for predictive analysis where we have 1 dependent and 1 independent variable.
- Good for forecasting effect, and determining the strength of the predictor.

**Huber Regression.**

- Huber regression is a regression technique that is robust to outliers.
- Uses a different loss function compared to the usual least square method.
- Seems usual to have the least squares penalty for little residuals of loss but on big residuals, its penalty is lesser and it linearly surges rather than quadratically.
- Hence, it works well for datasets having enormous outliers.

**Ridge Regression.**

- Ridge Regression analyses multiple regression data that has severe multicollinearity.
- Least squares method estimates are unbiased when multicollinearity occurs but their variances are huge that can mean that it may be far from the true value.
- Reduces the standard errors by adding a degree of bias to the regression estimates.

# CHAPTER

## 9. RESULTS AND DISCUSSIONS

The goal is to predict the quantity of change depending on the independent variables. These variables help in determining a trend between energy consumption, sales, and emissions within the manufacturing process using regression. During this study Linear, Huber, and Ridge Regression techniques are being implemented for a comparative study to know which works best during this case in terms of accuracy.

Only graphs and data heads of Huber regression have been shown in the paper to give readers an understanding of how the model works. This is to avoid repetition of results with similar figures. All three techniques provide similar results with very micro deviations which are not visible to the naked eye. This is because the accuracy of all the three techniques when applied for determining all the trends were approximately close enough 0.9995 to 0.9999 model scores.

**Sales Inference.**

The use of single use plastic has dominated the United Kingdom market since ages. Based on our data set we are trying to find the amount of gross expenditure done by companies in the UK annually. The predicted gross value is slightly lesser than the real gross value. This means that UK companies have started adhering to the Anti Plastic Pact and that is evident from the decrease in future average Gross proceeds. Companies will slowly start reducing their expenditure in manufacturing plastics. This means there will be a significant decrease in future production of single use plastics thus reducing the sale to a significant extent. Companies are yet to cut down on the usage of single use plastic. The UK government has presently enforced strict mechanisms by increasing the tax rate on plastics made of low-density polyethylene. This is visible through the reduction in future gross expenditure in graphs on Sales with VAT.

Over here Figure. 3. illustrates the actual gross expenditure and Figure. 4. talks about the predicted gross expenditure. Similarly, Figure. 5 represents the actual gross expenditure and Figure. 6. represents the predicted gross expenditure. In all the four graphs we notice that the actual gross expenditure and the predicted gross expenditure remains the same. This means the annual supply and sales of plastic in future will continue to keep increasing amongst the companies even after the government has enforced laws to reduce the usage of single use plastics.

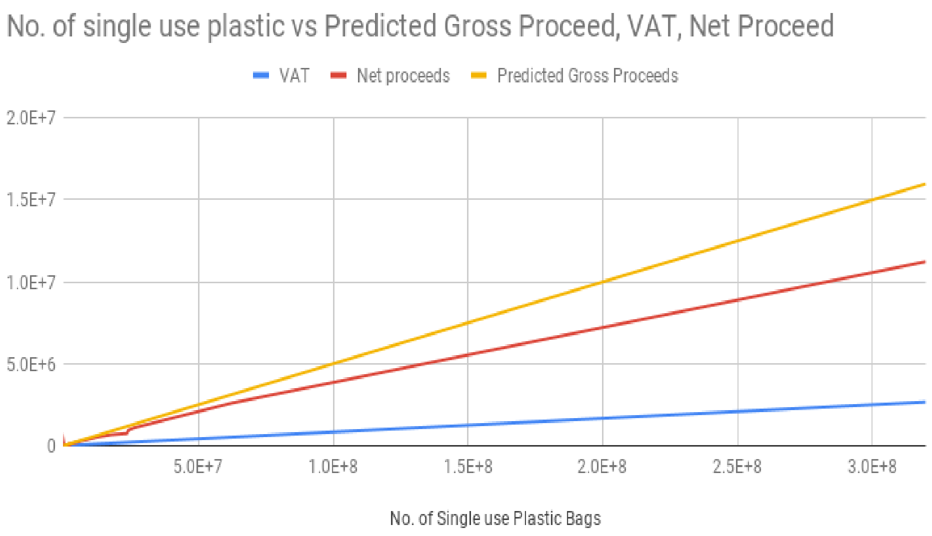



**Figure. 3.** No. of Single used Plastic vs Predicted Gross Proceed, VAT, Net Proceed in Huber

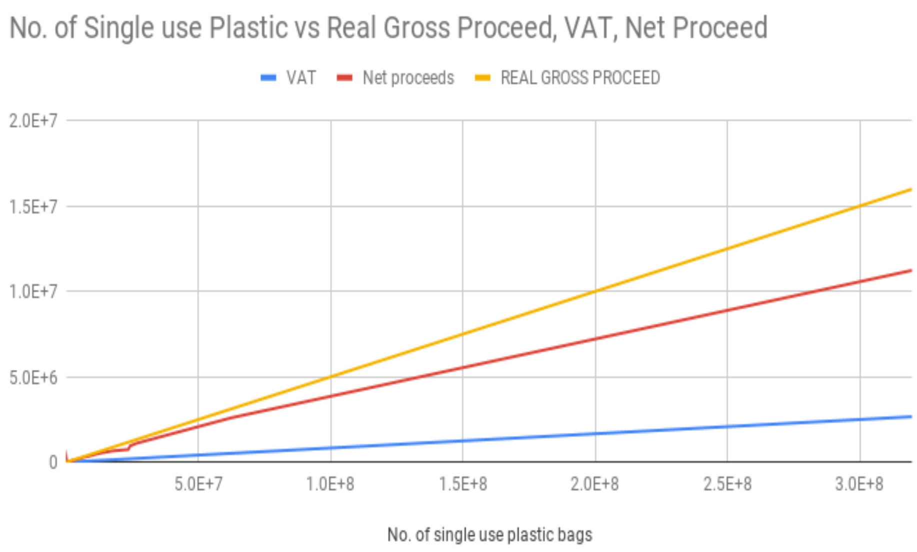



**Figure. 4.** No. of Single used Plastic vs Real Gross Proceed, VAT, Net Proceed in Huber

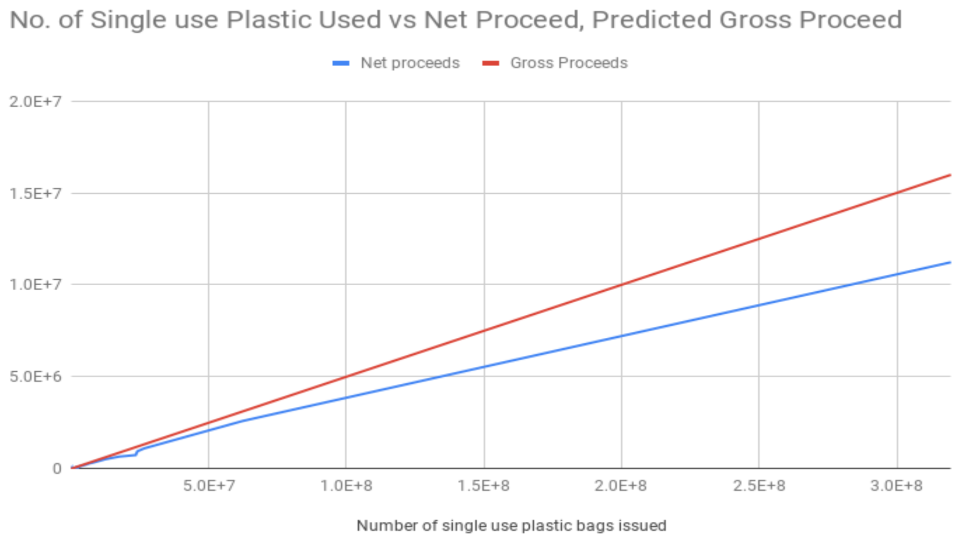



**Figure. 5**. No. of Single used Plastic vs Predicted Gross Proceed, Net Proceed in Huber

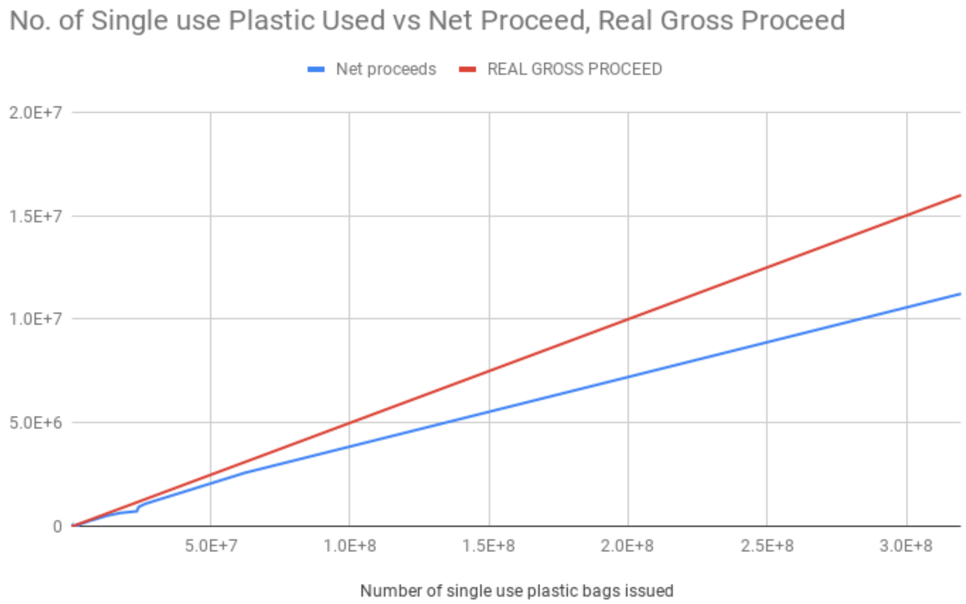



**Figure. 6.** No. of Single used Plastic vs Real Gross Proceed, Net Proceed in Huber

It is seen from the data set Figure. 7. and Figure. 8. that there is minute change in predicted gross which is nearly indistinguishable to the naked eye in the graph. Thus, the data set from Figure. 7 and Figure. 8 is compared to understand the minute change in the future gross expenditure. On comparing both the data it can be inferred that the data set with tax shows less future expenditure on single use plastic, while on the other hand the one without tax shows a linear increase in the future sales of plastic. This concludes that the government plays a huge role in controlling the production of single use of plastic. When the tax is increased, companies automatically reduce the consumption of plastic.

| No. of single use plastic bags | Net proceeds | Predicted Gross | Real Gross Proceed |
|---|---|---|---|
| 2 | 0 | 0.1 | 0 |
| 7 | 0 | 0.35 | 0 |
| 57 | 2 | 2.85 | 3 |
| 65 | 0 | 3.2497 | 3 |
| 108 | 5 | 5.4001 | 5 |
| 150 | 6 | 7.5 | 8 |
| 180 | 8 | 9.0001 | 9 |

**Figure. 7.** Head of top 7 values in Sales with VAT

| No. of single used plastics | VAT | Net proceeds | Predicted Gross | REAL GROSS PROCEED |
|---|---|---|---|---|
| 2 | 0 | 0 | 0.0999 | 0 |
| 7 | 0 | 0 | 0.3496 | 0 |
| 57 | 0 | 2 | 2.8488 | 3 |
| 65 | 0 | 0 | 3.2459 | 3 |
| 108 | 1 | 5 | 5.4008 | 5 |
| 150 | 2 | 6 | 7.5011 | 8 |
| 180 | 2 | 8 | 9.0016 | 9 |

**Figure. 8.** Head of top 7 values in Sales without VAT

All three regression techniques were tested for sales accuracy. The model Score of each of the regressions used in the sales phase are Ridge Regression: 0.9995181393016049, Huber Regression: 0.9999968840663866, Linear Regression: 0.9999275764740482. The model that gives the best accuracy is Huber followed by Linear and then Ridge. This is because it is less sensitive towards outliers or abnormalities in a data which is present in our data set.

**Material Phase Inference.**

Based on the graph Figure. 9. and Figure. 10., processing natural gas to prepare low density polyethylene consumes maximum energy. Petroleum comes second with total energy consumption. Remaining resources (coal, nuclear, others) is seen to consume very less energy. This clearly shows the excessive level of overconsumption of the presently two most depleting resources. Only the hubber regression graph is shown here. However, in all the 3 regressions, the predicted and actual results nearly coincide. This clearly shows the accuracy of the regression models. The amount of energy consumption for all resources continues to steeply increase clearly showing how much companies are over consuming non-renewable resources.

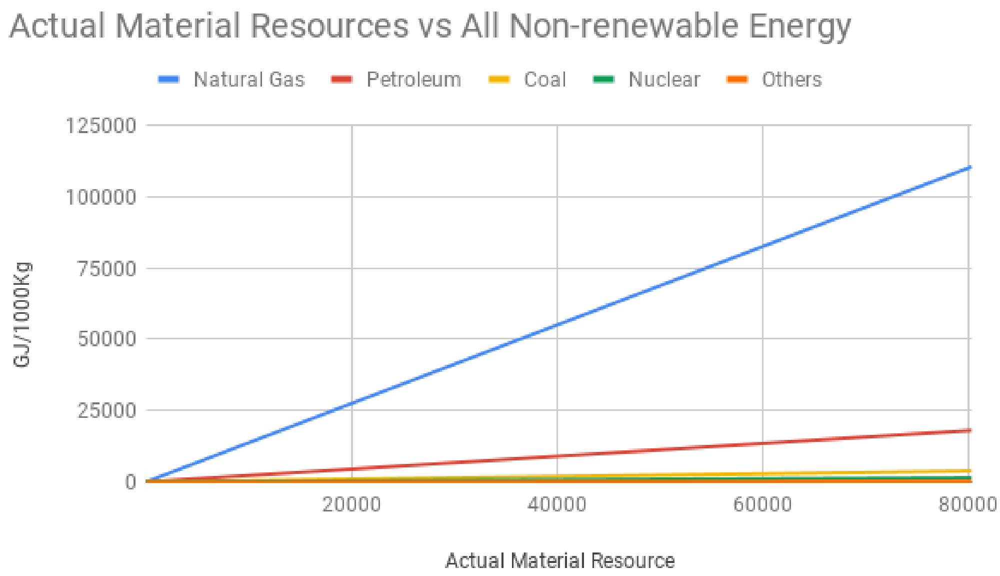



**Figure. 9.** Actual Material Resources vs All non-renewable Energy in Huber Regression

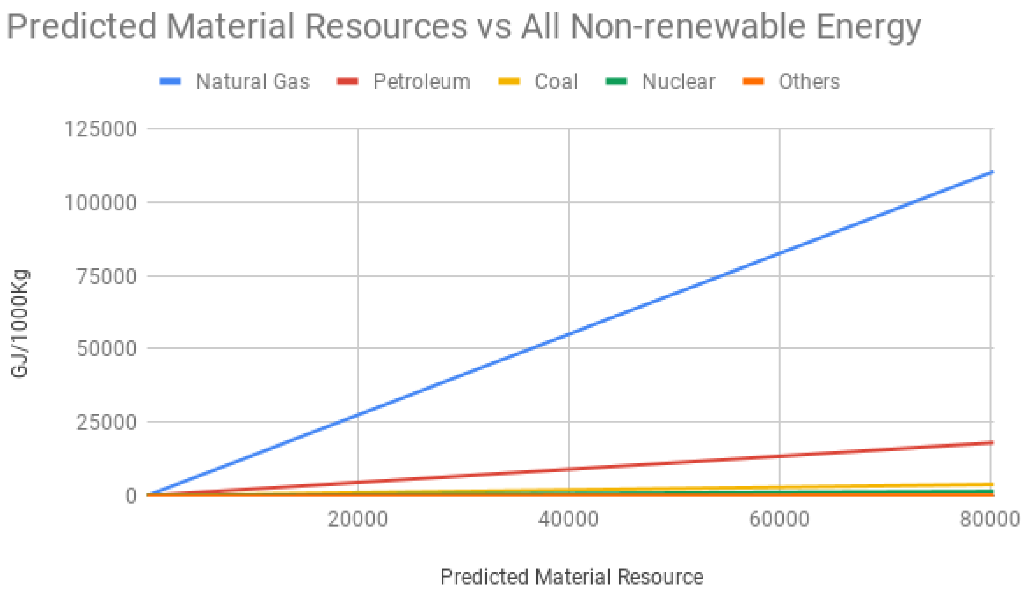



**Figure. 10.** Predicted Material Resources vs All non-renewable Energy in Huber Regression

**Process Phase Inference.**

The graph Figure. 11. depicts the comparison between actual energy consumption in the process phase and the graph Figure. 12. shows a comparison between the predicted energy consumption in the process phase. Natural gas consumes maximum energy to prepare low density polyethylene. Petroleum comes second with total energy consumption; other gasses consume very minimal energy. Over here the natural gas and petroleum is consumed in a similar manner to the material phase. The amount of energy consumption in the process phase is similar to that of the material phase. All the resources continue to steeply increase clearly showing how much companies are over consuming non-renewable resources.

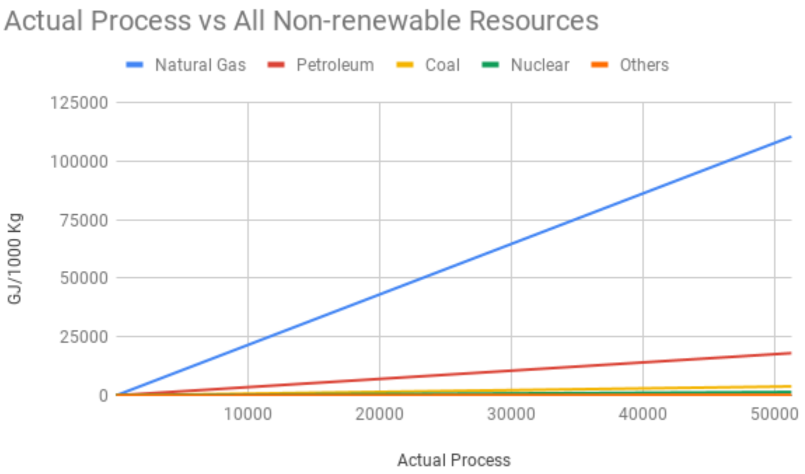



**Figure. 11.** Actual Process Resource vs All non-renewable Energy in Huber Regression

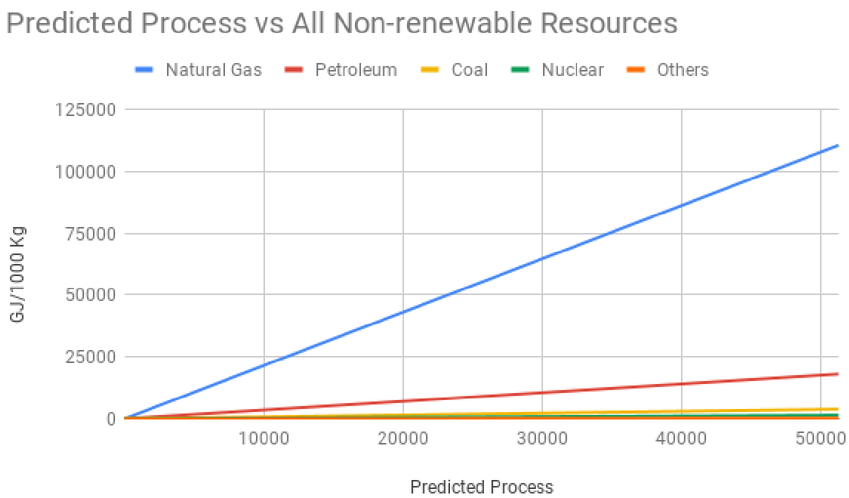



**Figure. 12.** Predicted Process Resource vs All non-renewable Energy in Huber Regression

**Transport Phase Inference.**

The graph Figure. 13. depicts the comparison between actual energy consumption in the transportation phase and the graph Figure. 14. shows a comparison between the predicted energy consumption in the transportation phase. The amount of energy consumption in the transportation phase is maximum while processing natural gas. Amount of petroleum used during transportation comes second in position with respect to energy consumption. There is a huge difference between the energy consumption of petroleum and natural gas. However, Coal, Nuclear and other non-renewable resources are required in a very minute amount. Petroleum and natural gas are also major sources of trapped methane gas. When these resources are over consumed, deadly methane is released into the air which is responsible for global warming.

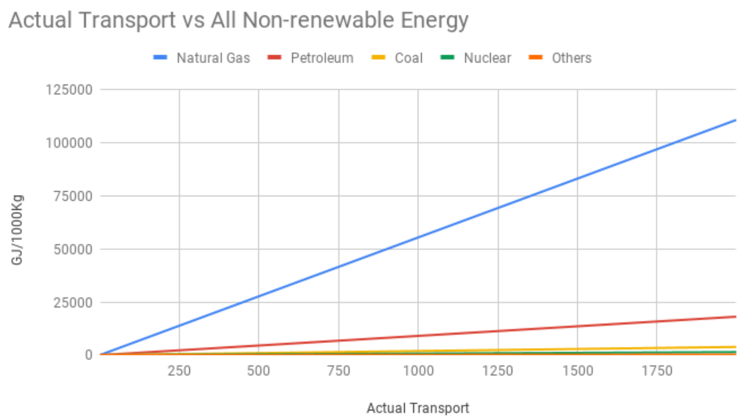



**Figure. 13.** Actual Transportation Resource vs All non-renewable Energy in Huber Regression

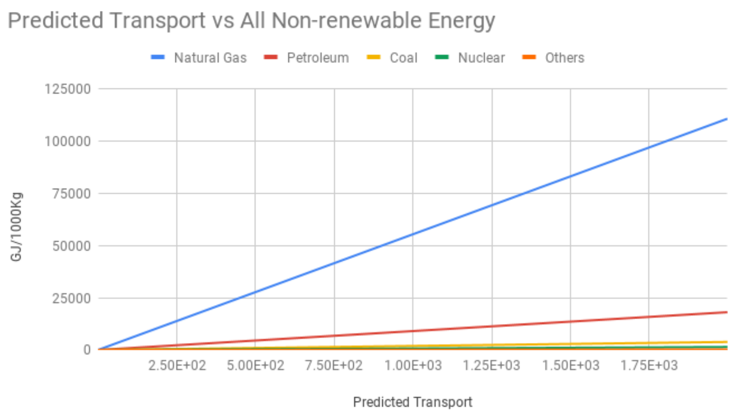



**Figure. 14.** Predicted Transportation Resource vs All non-renewable Energy in Huber Regression

**Combined Resource Inference.**

A comparison between non-renewable resources vs real and predicted values of process, transport and material resources were made. In the data head Figure. 15. consumption from resources is maximum for process value (the process during manufacturing of the plastic). In the second position comes the transportation phase (transportation of the raw materials to make the plastic). In the third position comes the extraction of raw materials to manufacture plastics.

The predicted value of the process phase is more than real values. This means the companies in the UK show an increase in the future trend of production of more plastic. The only way to reduce excessive production in future is by combining the manufacturing phase with sales of plastic. If the tax rate per production is increased then the demand will automatically decrease thus reducing excessive use of petroleum and natural gas. A complete opposite trend is noticed in the real value of the transportation phase and predicted value of transportation. Unlike the process phase where the real value is more than the predicted value. This means the rate of energy consumption in the transportation phase will decrease in future. The energy consumption in the material phase, both predicted and real value is nearly negligible in comparison to the production and transportation phase.

The data head shows the total energy consumption by process, transport and material phase. The greater the energy consumption, the greater is the depletion of natural resources. Thus, affecting the ecological factor of the planet. It is also increasing the rate of carbon dioxide and methane emission which is responsible for increasing the average temperature of the planet.

| A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|
| Natural Gas | Petroleum | Coal | Nuclear | Others | Predicted Process | Predicted Transportation | Predicted Material Resource | Real Process | Real Transport Values | Real Material Resource |
| 0.000692 | 0.000113 | 2.40E-05 | 8.90E-06 | 1.60E-06 | 0.0003210001266 | 1.25E-05 | 0.0005029988148 | 0.000321 | 0.000125 | 0.000503 |
| 0.002422 | 0.0003955 | 8.40E-05 | 3.12E-05 | 5.60E-06 | 0.001123500127 | 4.37E-05 | 0.001760498815 | 0.0011235 | 0.00004375 | 0.0017605 |
| 0.014532 | 0.002373 | 0.000504 | 0.0001869 | 3.36E-05 | 0.006741000127 | 0.0002624999177 | 0.01056299882 | 0.0091485 | 0.00035625 | 0.0143355 |
| 0.02249 | 0.0036725 | 0.00078 | 0.00028925 | 5.20E-05 | 0.01043250013 | 0.0004062499177 | 0.01634749882 | 0.0104325 | 0.00040625 | 0.0163475 |
| 0.023528 | 0.003842 | 0.000816 | 0.0003026 | 5.44E-05 | 0.01091400013 | 0.0004249999177 | 0.01710199882 | 0.017334 | 0.000675 | 0.027162 |
| 0.026988 | 0.004407 | 0.000936 | 0.0003471 | 6.24E-05 | 0.01251900013 | 0.0004874999177 | 0.01961699882 | 0.024075 | 0.0009375 | 0.037725 |
| 0.037368 | 0.006102 | 0.001296 | 0.0004806 | 8.64E-05 | 0.01733400013 | 0.0006749999177 | 0.02716199882 | 0.02889 | 0.001125 | 0.04527 |
| 0.038406 | 0.0062715 | 0.001332 | 0.00049395 | 8.88E-05 | 0.01781550013 | 0.0006937499177 | 0.02791649882 | 0.03531 | 0.001375 | 0.05533 |
| 0.0519 | 0.008475 | 0.0018 | 0.0006675 | 0.00012 | 0.02407500013 | 0.0009374999177 | 0.03772499882 | 0.038841 | 0.0015125 | 0.060863 |

**Figure. 15.** Head of top 9 values from combined Process, Transportation and Material Phase

**Greenhouse Emission Inference.**

In the previous graphs Figure. 9. – Figure. 14. we have noticed an excessive consumption of non-renewable resources. These resources when used for manufacturing the single using plastics often emit large amounts of harmful greenhouse gasses.

The Pie Chart in Figure. 16. analyses the amount of greenhouse gas that is predicted to be emitted when the single-use plastic is extracted, manufactured and transported. The greenhouse gasses that have been taken in consideration are Carbon Dioxide, Methane, Nitrous Oxide, Methyl Bromide, Methyl Chloride, Trichloroethane, Chloroform, Carbon Tetra Chloride, CFC13, HCFC 22. A pie chart has been used to represent all the pollutants

It is noticed that Carbon Dioxide and Methane are produced the most followed by other pollutants. Around 78.67% of Carbon Dioxide Fossil and 20.88% of Methane is emitted. This is because petroleum and natural gas resources are highly over consumed to manufacture single use plastics. When the non-renewable resources go into the manufacturing phase, $CO_2$ and methane are released as products that escape into the earth surface. During the transportation phase, a lot of coal petroleum and natural gas is used for fuel purposes. This releases harmful pollutants. Both methane and $CO_2$ act as a blanket that captures heat and doesn't allow them to escape from the surface. They make the air extremely hot thus responsible for major global warming and climate tipping points.

Methane and $CO_2$ stay for a very long time in the atmosphere. They are found trapped in abundance on the earth's surface. Methane is a deadlier greenhouse pollutant as it remains in the atmosphere for a longer time than $CO_2$. Both of them are significantly responsible for global warming. They can have a devastating effect on the average global temperature of the planet.

The steep rise in the curve for $CO_2$ and methane for all three phases show how it will not just cause a devastating effect on climate but also responsible for health issues like vision problems, memory loss, nausea, vomiting, facial flushing and headache

The remaining pollutants are produced in very micro amounts. They together constitute only 0.44% of the total pollution. So, if the emission rate is controlled for methane and CO2, then by default there is a possibility of the climatic tipping point from being controlled.

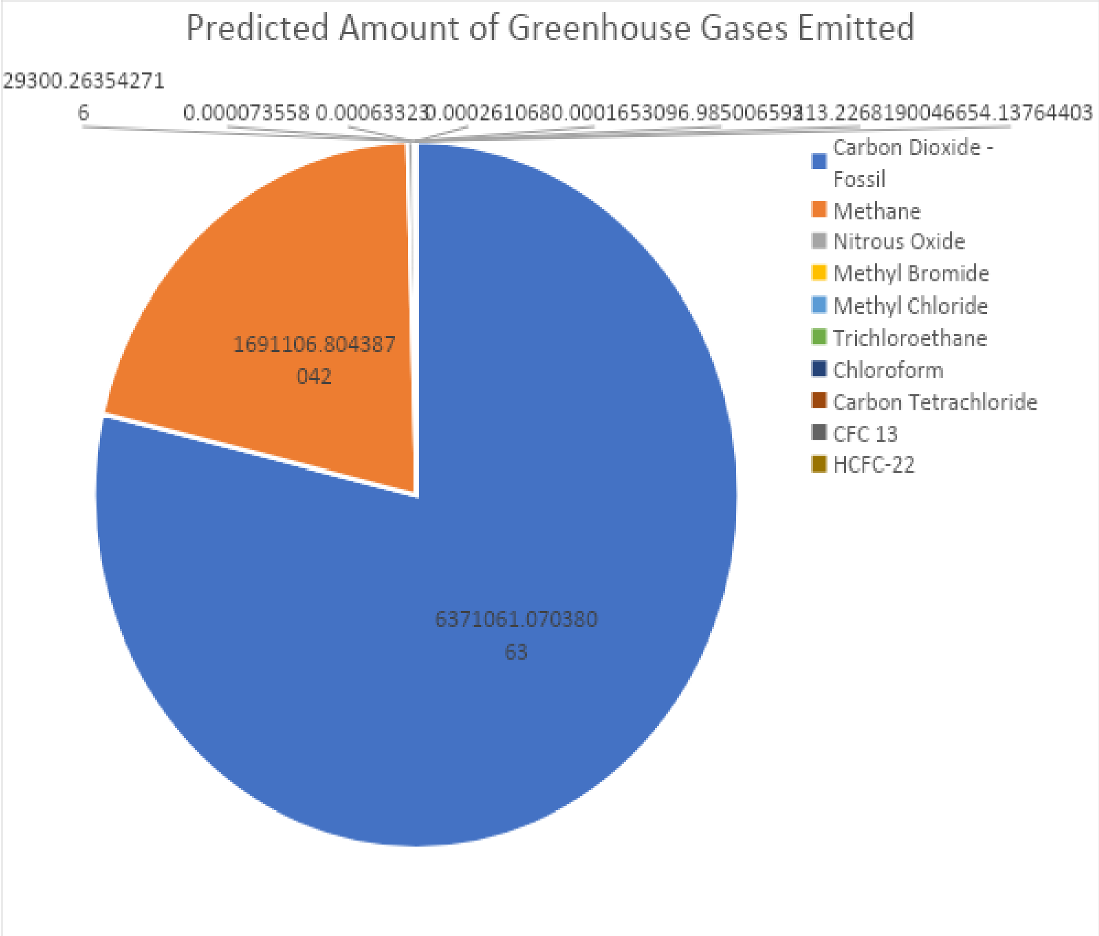



**Figure. 16.** Predicted Amount of Greenhouse Gases Emitted during all phases (in 1000 Kg)

**Non-Greenhouse Emission Inference.**

Similar Pie Chart Figure. 17. has been made to compare the levels of non-greenhouse gasses that are emitted during the process, transportation and material phase. These gases are not responsible for global warming but they are responsible for contamination of air causing deadly health issues.

The non-greenhouse pollutants taken into consideration are Particulate Matter (PM unspecified, 2.5, 10), VOC, CO2 non-fossil, nitrogen oxide, carbon monoxide and Sulphur dioxide. All three phases show the same trend.

It is observed that Sulphur dioxide is produced at a huge level (43.5 percent). This is an alarming matter as they not just damage the ecological system but also cause harm to plants. They are also responsible for major damage to heritage buildings by making them pale in colour. Such a large amount of SO2 may lead to issues such as shortness of breath and chest tightness

Nitrogen Oxides comes second in position with an emission percentage of 27 approx. They are majorly responsible for damaging the vegetation, animals and human life. Eye irritation, lung problem and breathing issue are just the start of the problem when Nitrogen Oxides is produced at such a large extent

Nitrogen Oxides along with Sulphur Dioxide which are emitted in abundance during the manufacturing of single use plastic form acid rain. Acid rain is very harmful to not just humans or plants but also to buildings and statues. They chemically react with the composition of these structures and dissolve the chemicals thus making them weak and brittle.

It may look like Carbon monoxide and VOC are produced at a decent rate when compared to Sulphur Dioxide and Nitrogen Oxides. However, 12 percent Carbon monoxide is still a moderate rate. This is safe because above 40 percent these pollutants become lethal to human and animal lives.

Particulate Matter (unspecified, 10 and 2.5) are emitted in very micro amounts in comparison to the other pollutants These are majorly micro and nano fine particles, which remain undetected to the human eye. These are possibly the main contributors to indoor pollution and they have a significant impact on human health. These are a complex mixture of extremely small particles and liquid droplets. Based on the particle size, Particulate Matter is categorized by PM 2.5 and PM10, which refer to particle sizes below 2.5 and 10 μm. These pollutants are deadly in nature and are responsible for even deaths when emitted at a huge level.

The emission from Carbon Dioxide fossil is almost negligible. They are less deadly than the above-mentioned pollutants but if produced in high concentration can lead to coma or death.

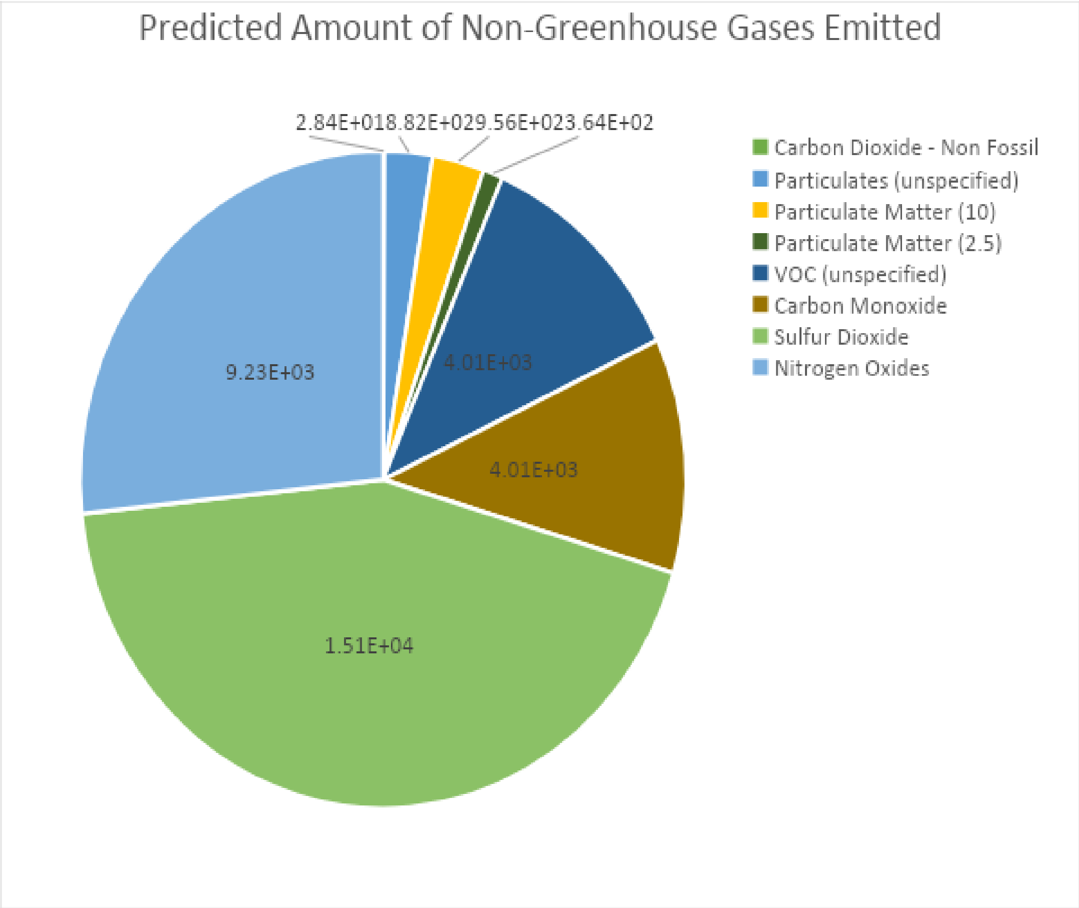



**Figure. 17.** Predicted Amount of Non-Greenhouse Gases Emitted during all phases (in 1000 Kg)

# CHAPTER

# 10. CONCLUSION

All three regression techniques were tested for accuracy. The model that gives the best accuracy is Huber followed by Linear and then Ridge. This is because it is less sensitive towards outliers or abnormalities in a data which is present in our data set. However, during the process certain values of net, gross and vat were not available, so they were assumed 0 by default. Availability of complete data has been an issue since ages and it was also reflected during this study. Most of these data are hidden or unexplored due to confidentiality reasons of companies. The data used in this process is independent on any specific location so the results might vary if any specific region is taken in consideration. The entire process using regression technique has provided with very high efficiency results. The readers could try clustering or classification technique as well to obtain a sustainable. result.

Recommendations for Future Work would be to find a better dataset which has not only the data about UK but also the entire European Nations or all Plastic Manufactures Data. This model was only tried out with regression but more advanced analytic methodologies could be used like Deep Learning.

**REFERENCES**

[1] Nieuwlaar, E. (2013). Life Cycle Assessment and Energy Systems. Reference Module in Earth Systems and Environmental Sciences, 3, 647–654. https://doi.org/10.1016/b978-0-12-409548-9.01334-8

[2] Guinée, J. B., Heijungs, R., Huppes, G., Zamagni, A., Masoni, P., Buonamici, R., Ekvall, & Rydberg, T. (2011). Life cycle assessment: Past, present, and future. Environmental Science and Technology, 45(1), 90–96. https://doi.org/10.1021/es101316v

[3] Murray PW, Agard B, Barajas MA. Forecasting supply chain demand by clustering customers. IFAC-PapersOnLine.2015;48(3):1834–9. https://doi.org/10.1016/J.IFACOL.2015.06.353.

[4] Pang Y, Yao B, Zhou X, Zhang Y, Xu Y, Tan Z. Hierarchical electricity time series forecasting for integrating consumption patterns analysis and aggregation consistency; 2018. In: IJCAI international joint conference on artificial intelligence; 2018, p. 3506

[5] Gaur M, Goel S, Jain E. Comparison between nearest Neighbours and Bayesian network for demand forecasting in supply chain management. In: 2015 international conference on computing for sustainable global development, INDIACom 2015, May; 2015, p. 1433–6.

[6] Merkuryeva G, Valberga A, Smirnov A. Demand forecasting in pharmaceutical supply chains: a case study. Procedia Comput Sci. 2019; 149:3–10. https://doi.org/10.1016/J.PROCS.2019.01.100.

[7] Yang CL, Sutrisno H. Short-term sales forecast of perishable goods for franchise business. In: 2018 10th international conference on knowledge and smart technology: cybernetics in the next decades, KST 2018, p. 101–5; 2018. https://doi.org/10.1109/KST.2018.8426091.

[8] M. Sarhani and A. El Afia, "Intelligent systembased support vector regression for supply chain demand forecasting," 2014 Second World Conference on Complex Systems (WCCS), Agadir, Morocco, 2014, pp. 79-83, doi: 10.1109/ICoCS.2014.7060941

[9] Guanghui, Wang. (2012). Demand Forecasting of Supply Chain Based on Support Vector Regression Method. Procedia Engineering. 29. 280–284. 10.1016/j.proeng.2011.12.707.

[10] Richey Jr, R.G., Morgan, T.R., Lindsey-Hall, K., Adams, F.G., 2016. A global exploration of Big Data in the supply chain. Int. J. Phys. Distrib. Logist. Manage. 46, 710–739.

[11] Kang, Y., Park, I., Rhee, J., Lee, Y., 2016. MongoDB-based repository design for IoT-generated RFID/sensor big data. IEEE Sens. J. 16, 485–497. http://dx.doi.org/10.1109/JSEN.2015.2483499

[12] Rehman, M.H., Chang, V., Batool, A., Wah, T.Y., 2016. Big data reduction framework for value creation in sustainable enterprises. Int. J. Inf. Manage. 36, 917–928. http://dx.doi.org/10.1016/j.ijinfomgt.2016.05.013

[13] Davenport, T.H., Harris, J.G., De Long, D.W., Jacobson, A.L., 2001. Data to knowledge to results: building an analytic capability. Calif. Manage. Rev. 43, 117–138. http://dx.doi.org/10.2307/41166078.

[14] Hazen, B.T., Boone, C.A., Ezell, J.D., Jones-Farmer, L.A., 2014. Data quality for data science, predictive analytics, and big data in supply chain management: an introduction to the problem and suggestions for research and applications. Int. J. Prod. Econ. 154, 72–80. http://dx.doi.org/10.1016/j.ijpe.2014.04.018.

[15] Wang, R.Y., Strong, D.M., 1996. Beyond accuracy: what data quality means to data consumers. J. Manage. Inf. Syst. 12, 5–33.

[16] Meixell, M., Wu, S., 2001. Scenario analysis of demand in a technology market using leading indicators. Semicond. Manuf. IEEE 14, 65–75.

[17] Blackburn, R., Lurz, K., Priese, B., Göb, R., Darkow, I.-L., 2015. A predictive analytics approach for demand forecasting in the process industry. Int. Trans. Oper. Res. 22, 407–428. http://dx.doi.org/10.1111/itor.12122.

[18] Dutta, D., Bose, I., 2015. Managing a big data project: the case of Ramco cements limited. Int. J. Prod. Econ. 165, 293–306. http://dx.doi.org/10.1016/j.ijpe.2014.12.032.

[19] Alfaro, L.A., Le, T.M.H., Choi, H.R., Cho, M.J., 2015. Deployment model of big data for port logistics. Inf. 18, 7–16.

[20] Radke, A.M., Tseng, M.M., 2015. Design considerations for building distributed supply chain management systems based on cloud computing. J. Manuf. Sci. Eng. 137, 40906. http://dx.doi.org/10.1115/1.4030736.

[21] Sanders, N.R., 2016. How to use big data to drive your supply chain. Calif. Manage. Rev. 58, 26–48. http://dx.doi.org/10.1525/cmr.2016.58.3.26.

[22] Zhichao Liu, Qiuhong Jiang, Zhang, H. (2012). Life Cycle Assessment Model: Recent Development and Future Tendency. Proceedings of the 10th International Conference on Frontiers of Design and Manufacturing, June.

[23] Park, J. H., & Seo, K.-K. (2003). Approximate life cycle assessment of product concepts using multiple regression analysis and artificial neural networks. KSME International Journal, 17(12), 1969–1976. https://doi.org/10.1007/bf02982436

[24] Carnegie Mellon University. (2016). Economic Input-Output Life Cycle Assessment - Carnegie Mellon University. http://www.eiolca.net/

[25] Matsumoto, H. (n.d.). System Dynamics Model For Life Cycle Assessment (LCA) Of Residential Buildings Hiroshi Matsumoto Department of Architecture and Civil Engineering , Toyohashi University of Technology Toyohashi 441-8580 , Japan. Energy.

[26] Nuasaen, S., Opaprakasit, P., & Tangboriboonrat, P. (2014). Hollow latex particles functionalized with chitosan for the removal of formaldehyde from indoor air. Carbohydrate Polymers, 101(1), 179– 187. https://doi.org/10.1016/j.carbpol.2013.09.059

[27] Moriguchi, Y., & Terazono, A. (2000). A simplified model for spatially differentiated impact assessment of air emissions. International Journal of Life Cycle Assessment, 5(5), 281–286. https://doi.org/10.1007/BF02977580

[28] Hochschorner, E., & Finnveden, G. (2003). Evaluation of two simplified life cycle assessment methods. International Journal of Life Cycle Assessment, 8(3), 119–128. https://doi.org/10.1007/BF02978456

[29] Park, J. H., & Seo, K. K. (2006). A knowledgebased approximate life cycle assessment system for evaluating environmental impacts of product design alternatives in a collaborative design environment. Advanced Engineering Informatics, 20(2), 147– 154. https://doi.org/10.1016/j.aei.2005.09.003

[30] Eisenhard, J. L., & De Schepper, M. S. (2000). Approximate Life-Cycle Assessment in Conceptual Product Design. ASME Design Engineering Technical Conferences & Computers and Information in Engineering Conference.

[31] Chen, J. L., & Liau, C. W. (2001). A simple life cycle assessment method for green product conceptual design. Proceedings - 2nd International Symposium on Environmentally Conscious Design and Inverse Manufacturing, 775–780. https://doi.org/10.1109/ECODIM.2001.992466

[32] Ayoub, N., Musharavati, F., Pokharel, S., & Gabbar, H. A. (2015). Risk based life cycle assessment conceptual framework for energy supply systems in large buildings. Journal of Cleaner Production, 107, 291–309. https://doi.org/10.1016/j.jclepro.2015.04.075

[33] Brondi, C., & Carpanzano, E. (2011). A modular framework for the LCA-based simulation of production systems. CIRP Journal of Manufacturing Science and Technology, 4(3), 305–312. https://doi.org/10.1016/j.cirpj.2011.06.006

[34] Fornasiero, R., Brondi, C., & Collatina, D. (2017). Proposing an integrated LCA-SCM model to evaluate the sustainability of customisation strategies. International Journal of Computer Integrated Manufacturing, 00(00), 1–14. https://doi.org/10.1080/0951192X.2016.1268716

[35] Cohas, E., Hendrickson, C., Lave, L., & Mcmichaell, F. (1995). Economic Input / Output Analysis to Aid Life Cycle Assessment of Electronics Products1 by 2. Environmental Impacts of Electronics Product Manufacture.

[36] Barrett, W. M., Strunjaš-yoshikawa, S., & Bell, J. H. (2007). Extension of Computer Aided Process Engineering Applications to Environmental Life Cycle Assessment and Supply Chain Management. 1187–1192.

[37] Suparno Bhatta, et. al., Arriving at Optimal Cost for a Sustainable Product using an Integrated LCA-SCM Model, Journal of Information Technology in Industry, Vol. 9, No. 2, 2021, pp 1193-1199, doi: https://doi.org/10.17762/itii.v9i2.472

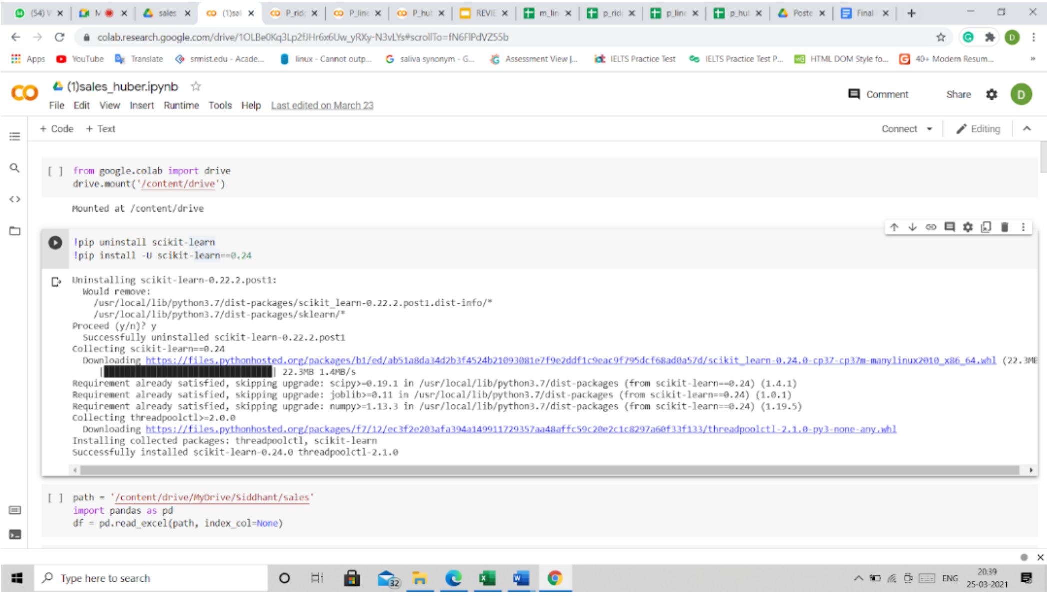
[38] Rosanna Fornasieroa, Carlo Brondia and Davide Collatinaa, International Journal of Computer Integrated Manufacturing, 2017

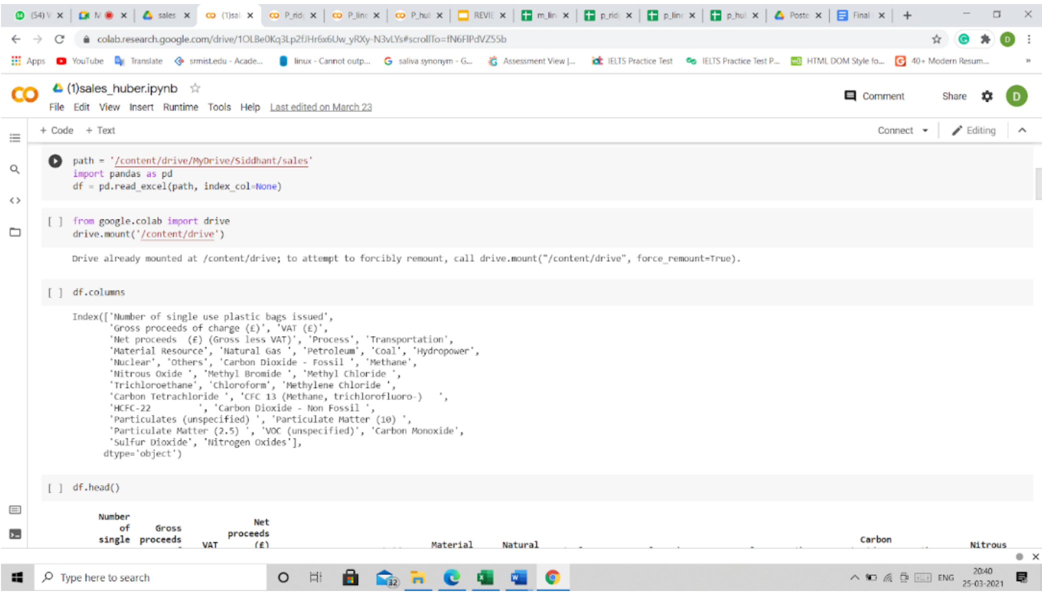
[39] William M. Barrett, Svetlana Strunjaš-Yoshikawa, Jonathan H Bell, 17th European Symposium on Computer Aided Process Engineering – ESCAPE17.

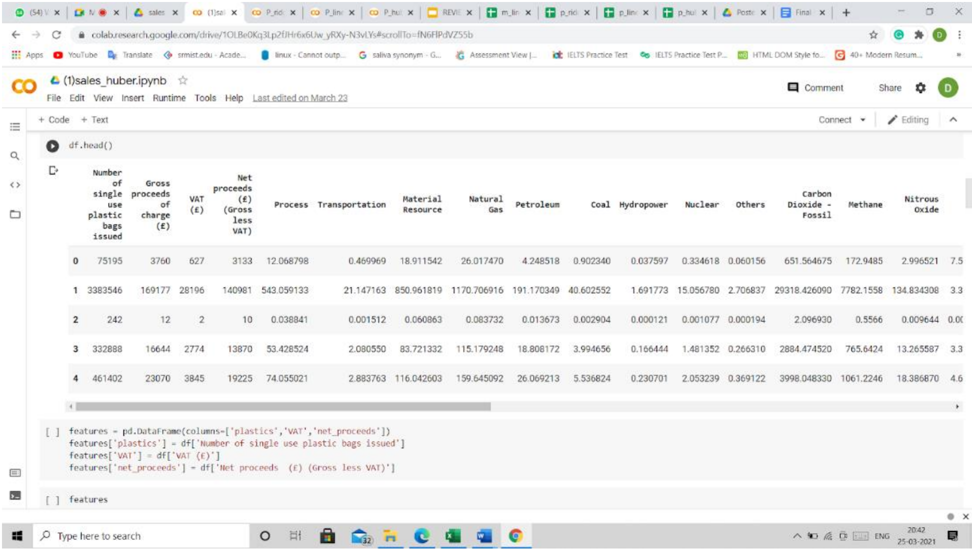
[40] **Dataset:** https://ckan.publishing.service.gov.uk/dataset/single-use-plastic-carrier-bags-charge-data-for-england.

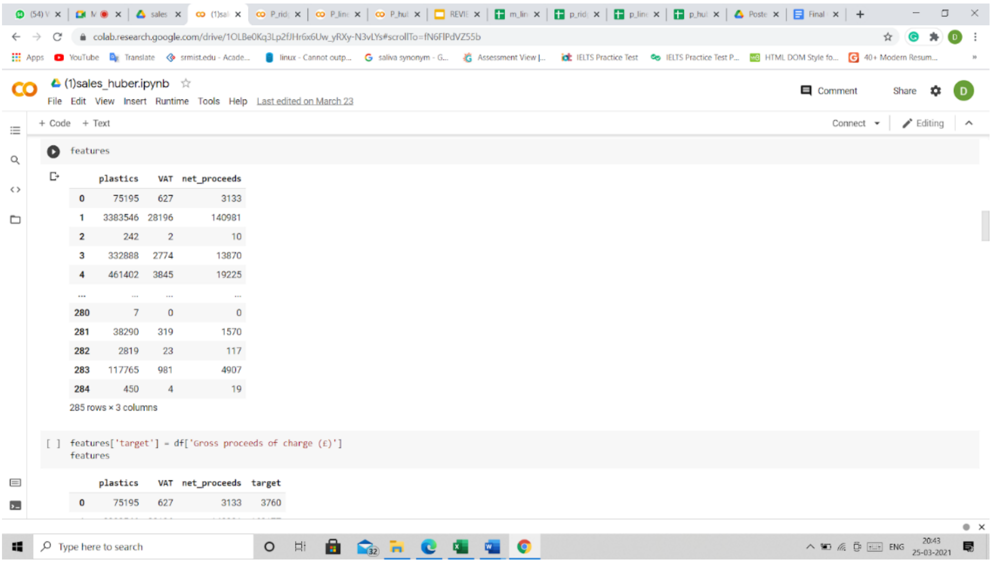
# APPENDIX 1 – SCREENSHOTS

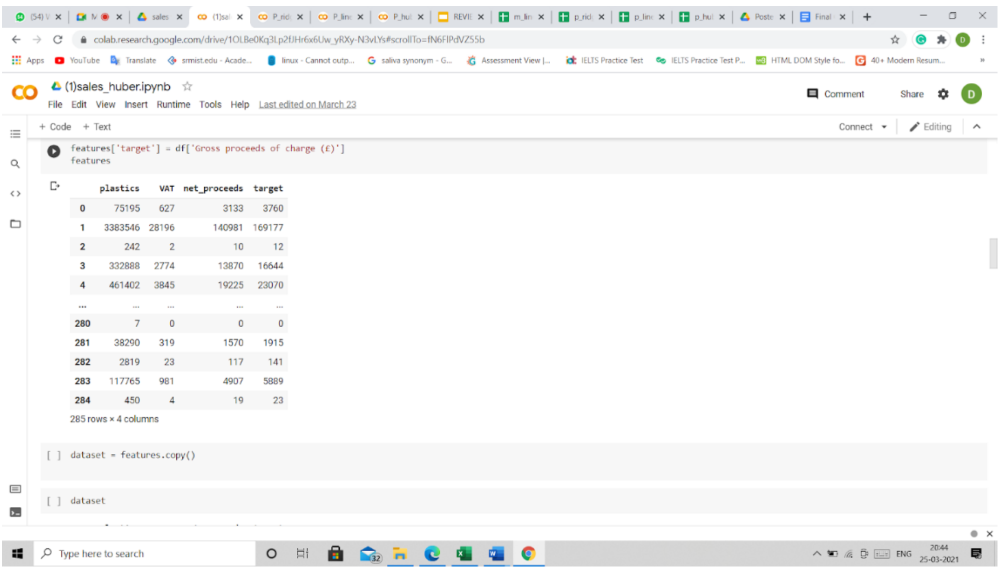
Here are the screenshots of the project that was implemented in Google Colab.

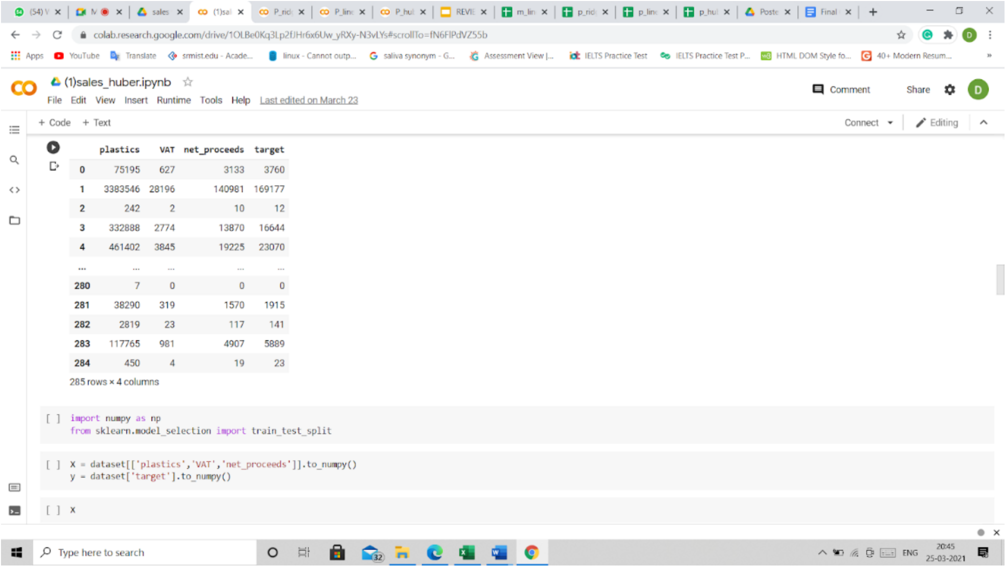





(1)sales_huber.ipynb
df.head()
Number of single use plastic bags issued | Gross proceeds of charge (£) | VAT (£) | Net proceeds (£) (Gross less VAT) | Process | Transportation | Material Resource | Natural Gas | Petroleum | Coal | Hydropower | Nuclear | Others | Carbon Dioxide - Fossil | Methane | Nitrous Oxide
0 75195 3760 627 3133 12.068798 0.469969 18.911542 26.017470 4.248518 0.902340 0.037597 0.334618 0.060156 651.564675 172.9485 2.996521 7.5
1 3383546 169177 28196 140981 543.059133 21.147163 850.961819 1170.706916 191.170349 40.602552 1.691773 15.056780 2.706837 29318.426090 7782.1558 134.834308 3.3
2 242 12 2 10 0.038841 0.001512 0.060863 0.083732 0.013673 0.002904 0.000121 0.001077 0.000194 2.096930 0.5566 0.009644 0.0(
3 332888 16644 2774 13870 53.428524 2.080550 83.721332 115.179248 18.808172 3.994656 0.166444 1.481352 0.266310 2884.474520 765.6424 13.265587 3.3
4 461402 23070 3845 19225 74.055021 2.883763 116.042603 159.645092 26.069213 5.536824 0.230701 2.053239 0.369122 3998.048330 1061.2246 18.386870 4.6
features = pd.DataFrame(columns=['plastics','VAT','net_proceeds'])
features['plastics'] = df['Number of single use plastic bags issued']
features['VAT'] = df['VAT (£)']
features['net_proceeds'] = df['Net proceeds (£) (Gross less VAT)']
features


(1)sales_huber.ipynb
features
plastics VAT net_proceeds
0 75195 627 3133
1 3383546 28196 140981
2 242 2 10
3 332888 2774 13870
4 461402 3845 19225
... ... ... ...
280 7 0 0
281 38290 319 1570
282 2819 23 117
283 117765 981 4907
284 450 4 19
285 rows × 3 columns
features['target'] = df['Gross proceeds of charge (£)']
features
plastics VAT net_proceeds target
0 75195 627 3133 3760


(1)sales_huber.ipynb
features['target'] = df['Gross proceeds of charge (£)']
features
plastics VAT net_proceeds target
0 75195 627 3133 3760
1 3383546 28196 140981 169177
2 242 2 10 12
3 332888 2774 13870 16644
4 461402 3845 19225 23070
... ... ... ... ...
280 7 0 0 0
281 38290 319 1570 1915
282 2819 23 117 141
283 117765 981 4907 5889
284 450 4 19 23
285 rows × 4 columns
dataset = features.copy()
dataset



(1)sales_huber.ipynb

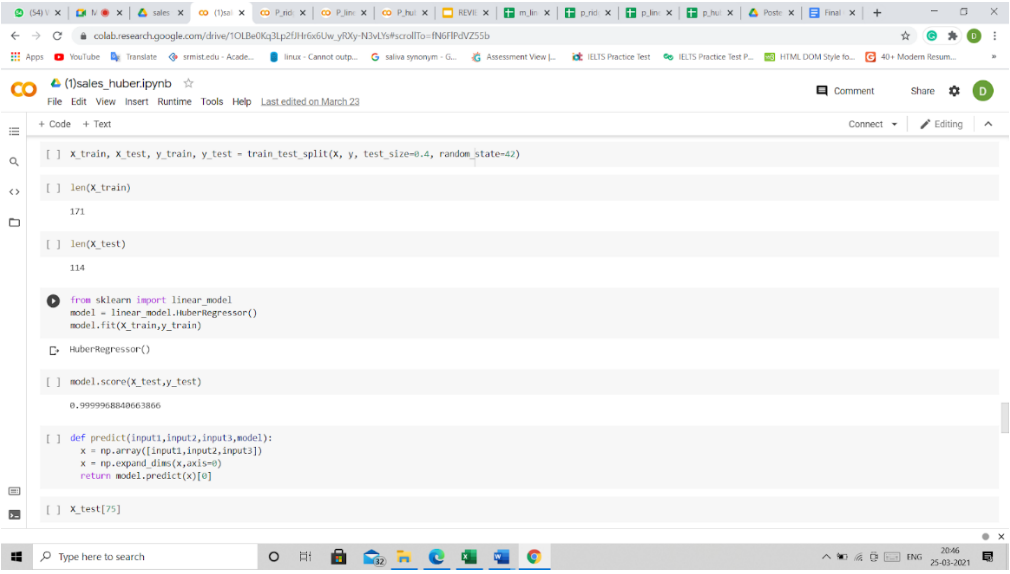
| | plastics | VAT | net_proceeds | target |
|---|---|---|---|---|
| 0 | 75195 | 627 | 3133 | 3760 |
| 1 | 3383546 | 28196 | 140981 | 169177 |
| 2 | 242 | 2 | 10 | 12 |
| 3 | 332888 | 2774 | 13870 | 16644 |
| 4 | 461402 | 3845 | 19225 | 23070 |
| ... | ... | ... | ... | ... |
| 280 | 7 | 0 | 0 | 0 |
| 281 | 38290 | 319 | 1570 | 1915 |
| 282 | 2819 | 23 | 117 | 141 |
| 283 | 117765 | 981 | 4907 | 5889 |
| 284 | 450 | 4 | 19 | 23 |

285 rows × 4 columns

```
import numpy as np
from sklearn.model_selection import train_test_split
```

```
X = dataset[['plastics','VAT','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()
```

```
X
```



(1)sales_huber.ipynb

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

```
len(X_train)
```

171

```
len(X_test)
```

114

```
from sklearn import linear_model
model = linear_model.HuberRegressor()
model.fit(X_train,y_train)
```

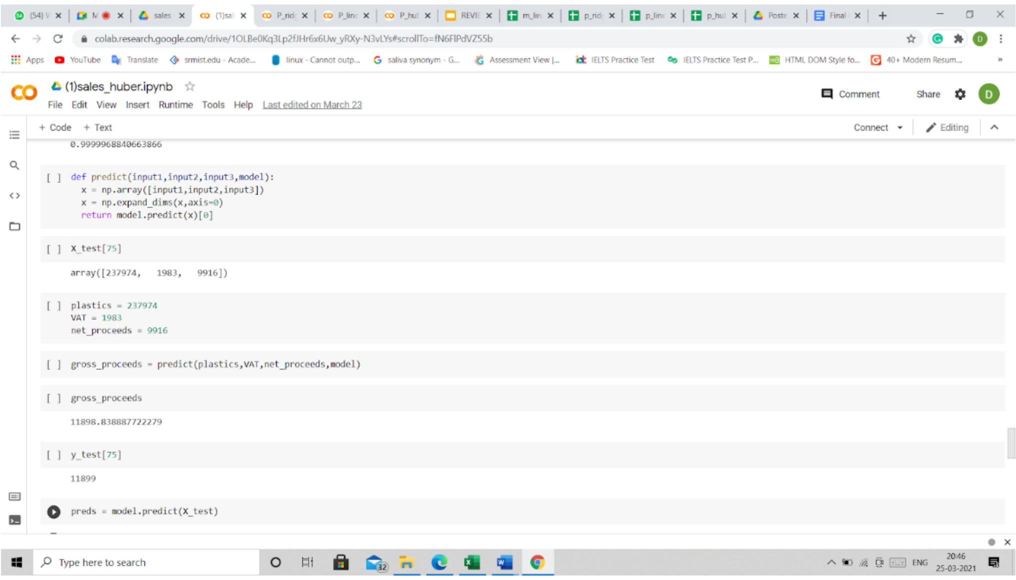
HuberRegressor()

```
model.score(X_test,y_test)
```

0.9999968840663866

```
def predict(input1,input2,input3,model):
  x = np.array([input1,input2,input3])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]
```

```
X_test[75]
```



(1)sales_huber.ipynb

0.9999968840663866

```
def predict(input1,input2,input3,model):
  x = np.array([input1,input2,input3])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]
```

```
X_test[75]
```

array([237974,   1983,   9916])

```
plastics = 237974
VAT = 1983
net_proceeds = 9916
```

```
gross_proceeds = predict(plastics,VAT,net_proceeds,model)
```

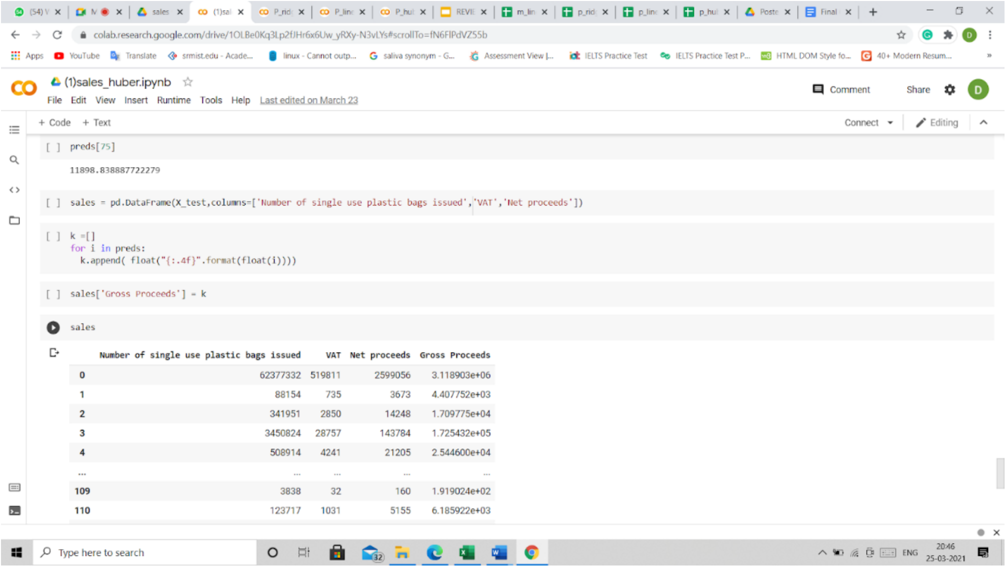
```
gross_proceeds
```

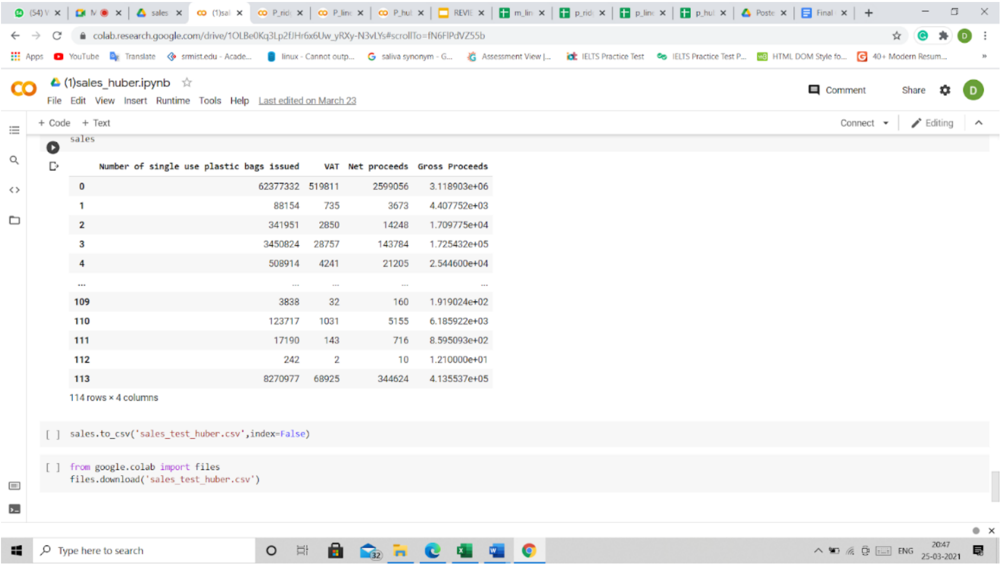
11898.838887722279

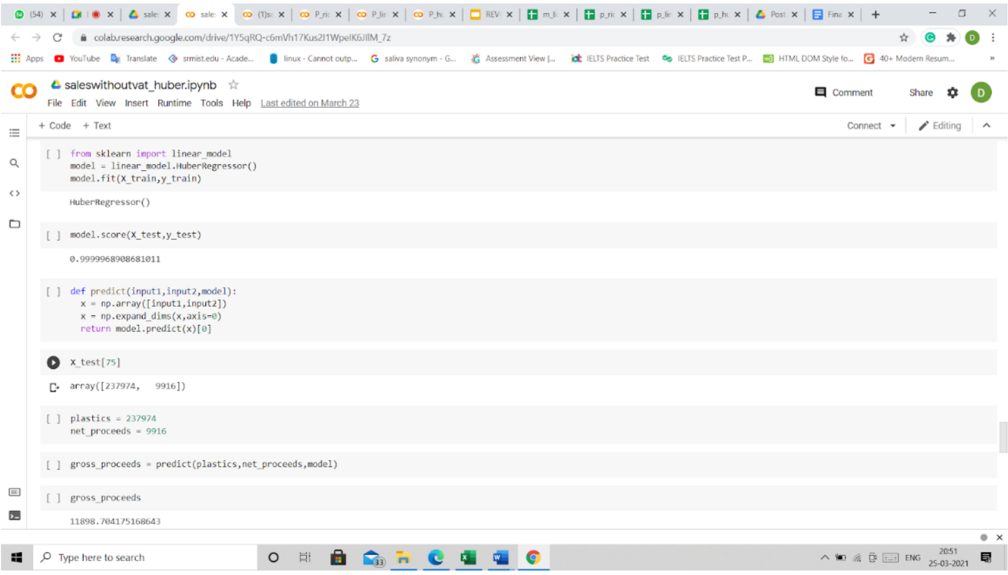
```
y_test[75]
```

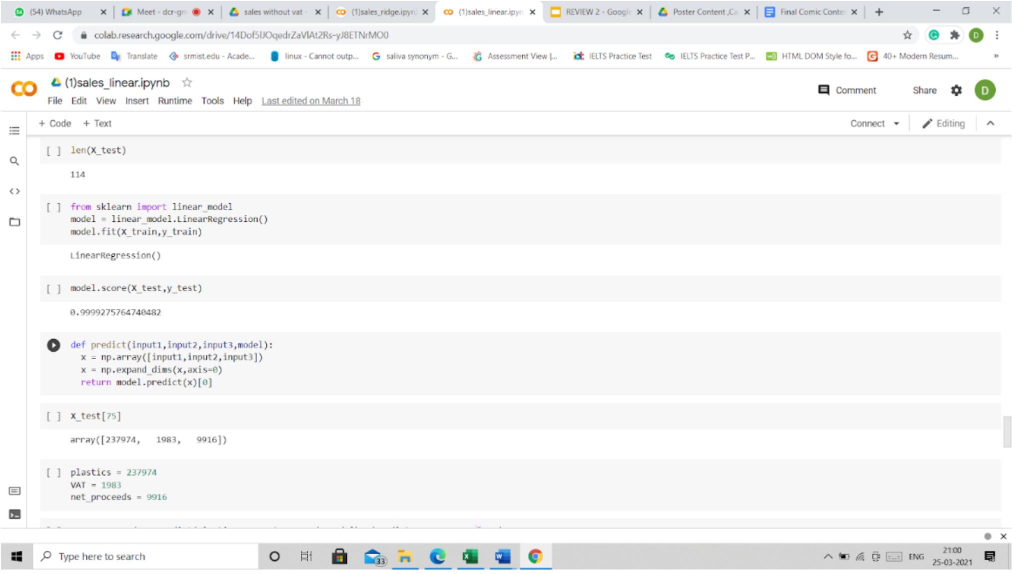
11899

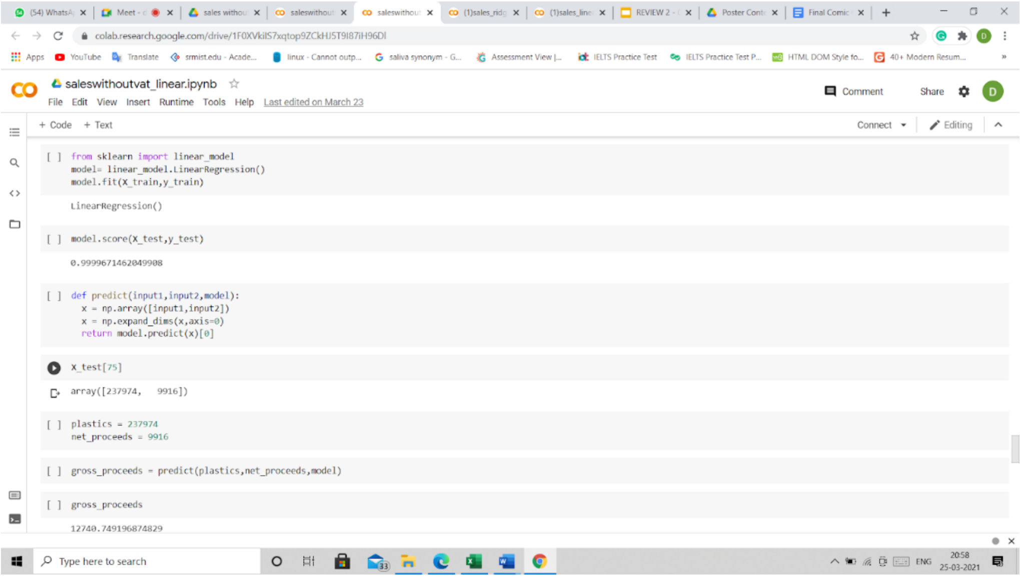
```
preds = model.predict(X_test)
```

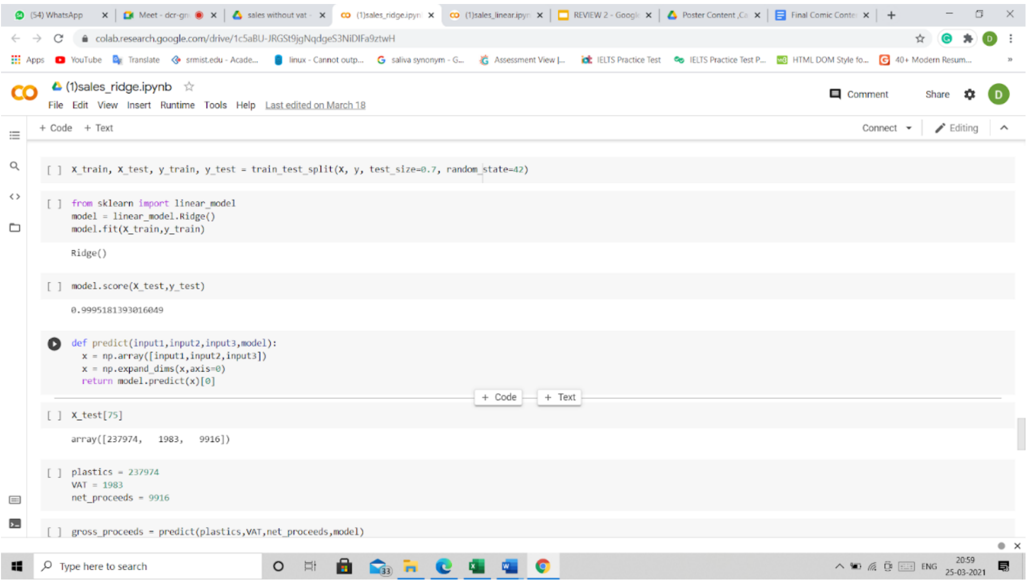

(1)sales_huber.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 23
preds[75]
11898.838887722279
sales = pd.DataFrame(X_test,columns=['Number of single use plastic bags issued','VAT','Net proceeds'])
k =[]
for i in preds:
  k.append( float("{:.4f}".format(float(i))))
sales['Gross Proceeds'] = k
sales
Number of single use plastic bags issued VAT Net proceeds Gross Proceeds
0 62377332 519811 2599056 3.118903e+06
1 88154 735 3673 4.407752e+03
2 341951 2850 14248 1.709775e+04
3 3450824 28757 143784 1.725432e+05
4 508914 4241 21205 2.544600e+04
... ... ... ... ...
109 3838 32 160 1.919024e+02
110 123717 1031 5155 6.185922e+03

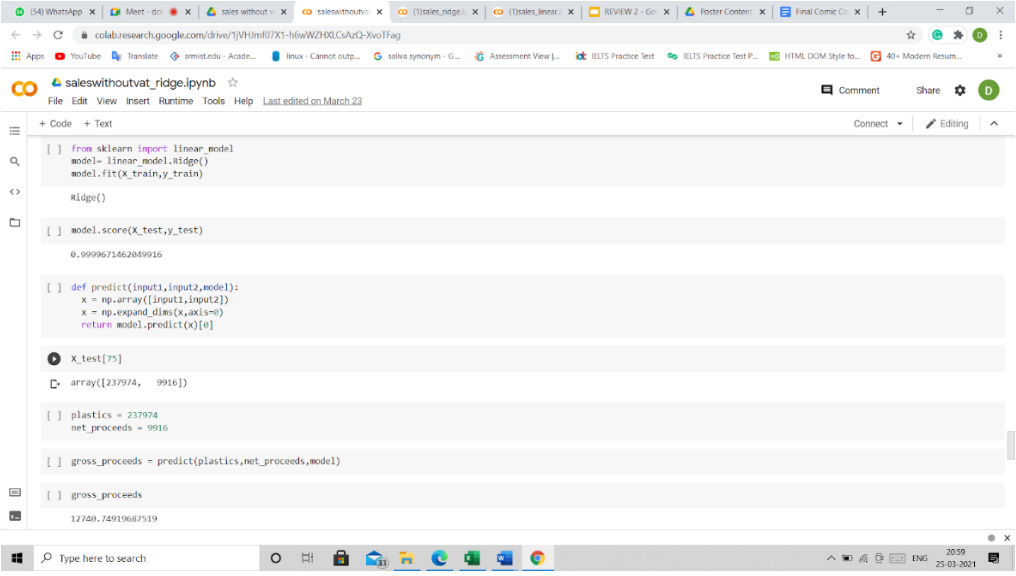


(1)sales_huber.ipynb
sales
Number of single use plastic bags issued VAT Net proceeds Gross Proceeds
0 62377332 519811 2599056 3.118903e+06
1 88154 735 3673 4.407752e+03
2 341951 2850 14248 1.709775e+04
3 3450824 28757 143784 1.725432e+05
4 508914 4241 21205 2.544600e+04
... ... ... ... ...
109 3838 32 160 1.919024e+02
110 123717 1031 5155 6.185922e+03
111 17190 143 716 8.595093e+02
112 242 2 10 1.210000e+01
113 8270977 68925 344624 4.135537e+05
114 rows × 4 columns
sales.to_csv('sales_test_huber.csv',index=False)
from google.colab import files
files.download('sales_test_huber.csv')

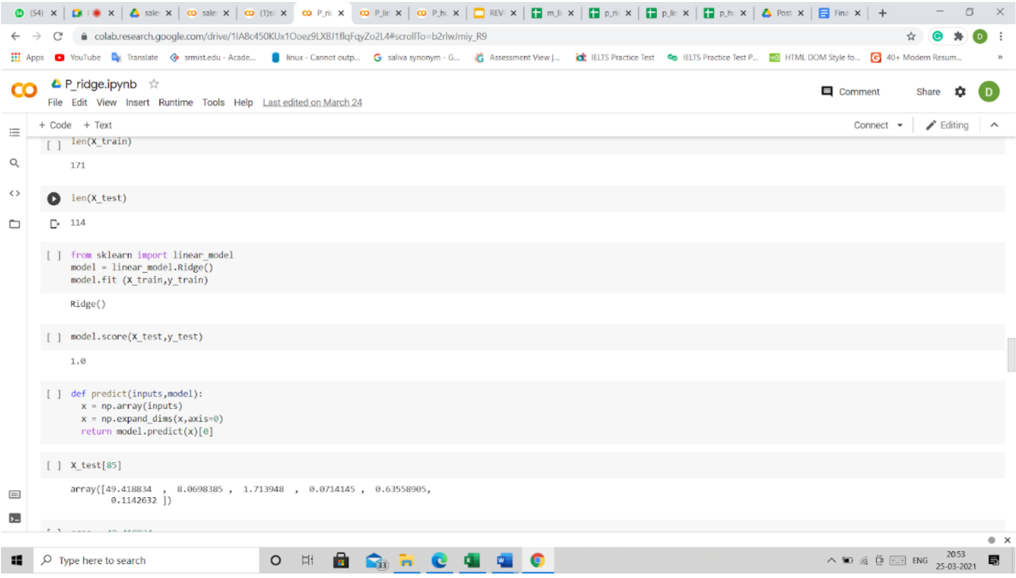


saleswithoutvat_huber.ipynb
from sklearn import linear_model
model = linear_model.HuberRegressor()
model.fit(X_train,y_train)
HuberRegressor()
model.score(X_test,y_test)
0.9999968908681011
def predict(input1,input2,model):
  x = np.array([input1,input2])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]
X_test[75]
array([237974,    9916])
plastics = 237974
net_proceeds = 9916
gross_proceeds = predict(plastics,net_proceeds,model)
gross_proceeds
11898.704175168643

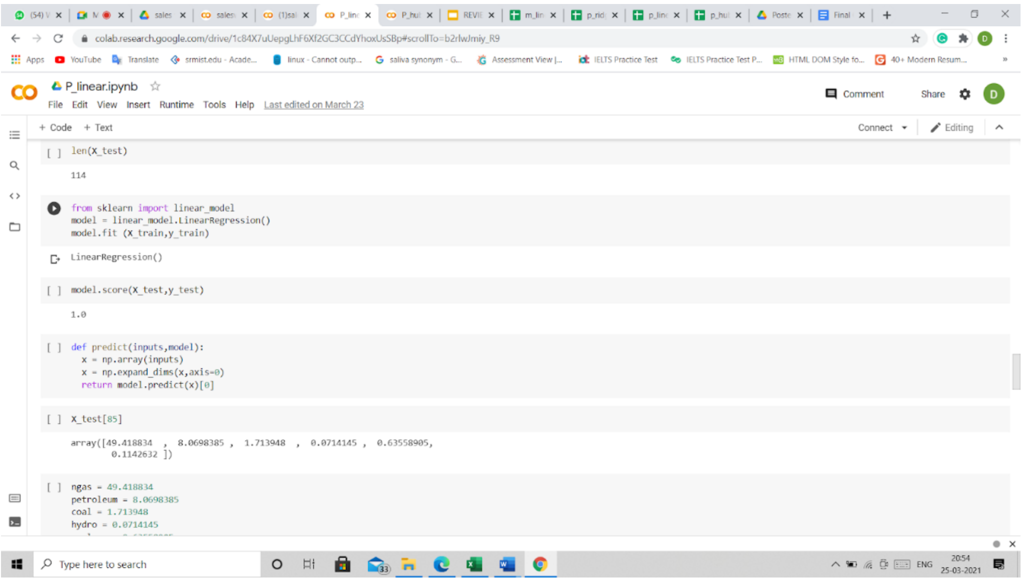

(1)sales_linear.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 18
len(X_test)
114
from sklearn import linear_model
model = linear_model.LinearRegression()
model.fit(X_train,y_train)
LinearRegression()
model.score(X_test,y_test)
0.9999275764740482
def predict(input1,input2,input3,model):
x = np.array([input1,input2,input3])
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
x_test[75]
array([237974, 1983, 9916])
plastics = 237974
VAT = 1983
net_proceeds = 9916

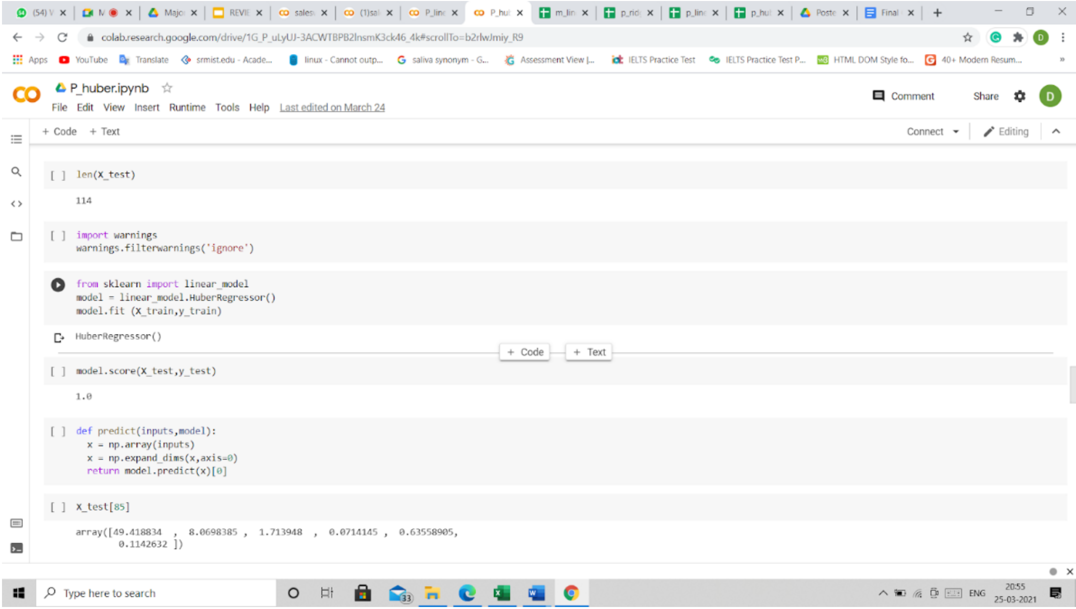

saleswithoutvat_linear.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 23
from sklearn import linear_model
model= linear_model.LinearRegression()
model.fit(X_train,y_train)
LinearRegression()
model.score(X_test,y_test)
0.9999671462049908
def predict(input1,input2,model):
x = np.array([input1,input2])
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
x_test[75]
array([237974, 9916])
plastics = 237974
net_proceeds = 9916
gross_proceeds = predict(plastics,net_proceeds,model)
gross_proceeds
12740.749196874829

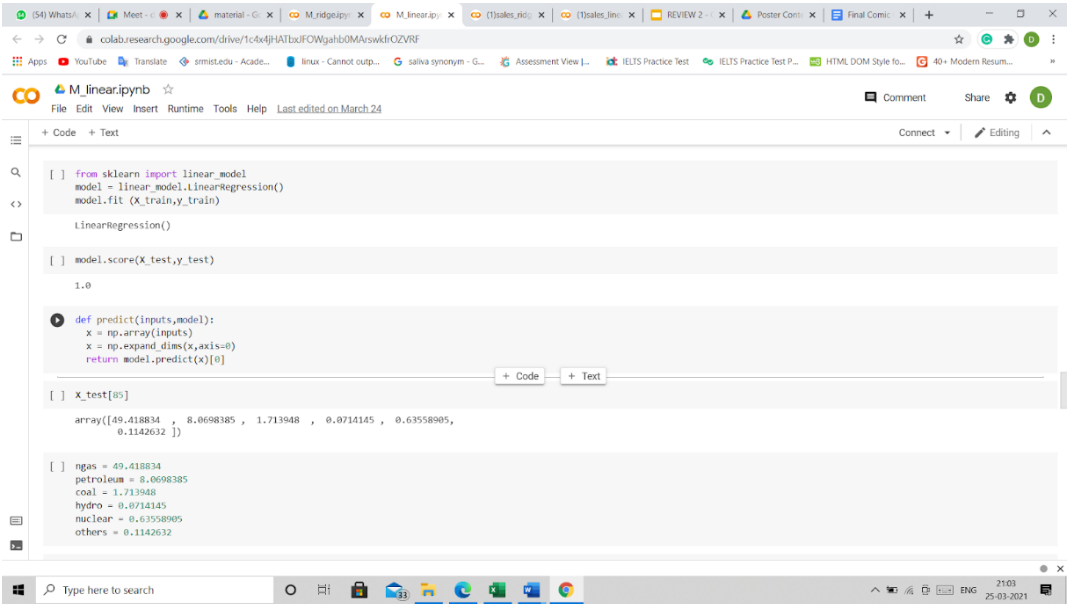

(1)sales_ridge.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 18
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.7, random_state=42)
from sklearn import linear_model
model = linear_model.Ridge()
model.fit(X_train,y_train)
Ridge()
model.score(X_test,y_test)
0.9995181393016049
def predict(input1,input2,input3,model):
x = np.array([input1,input2,input3])
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
x_test[75]
array([237974, 1983, 9916])
plastics = 237974
VAT = 1983
net_proceeds = 9916
gross_proceeds = predict(plastics,VAT,net_proceeds,model)

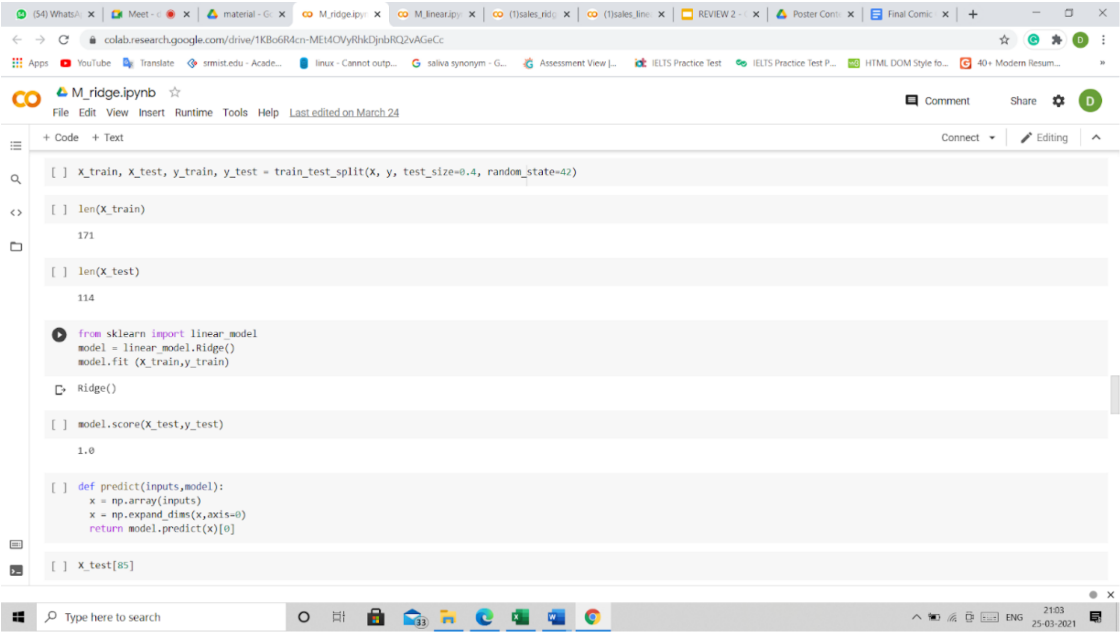

saleswithoutvat_ridge.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 23
from sklearn import linear_model
model= linear_model.Ridge()
model.fit(X_train,y_train)
Ridge()
model.score(X_test,y_test)
0.9999671462049916
def predict(input1,input2,model):
x = np.array([input1,input2])
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
X_test[75]
array([237974, 9916])
plastics = 237974
net_proceeds = 9916
gross_proceeds = predict(plastics,net_proceeds,model)
gross_proceeds
12740.74919687519


P_ridge.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 24
len(X_train)
171
len(X_test)
114
from sklearn import linear_model
model = linear_model.Ridge()
model.fit (X_train,y_train)
Ridge()
model.score(X_test,y_test)
1.0
def predict(inputs,model):
x = np.array(inputs)
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
X_test[85]
array([49.418834 , 8.0698385 , 1.713948 , 0.0714145 , 0.63558905,
0.1142632 ])


P_linear.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 23
len(X_test)
114
from sklearn import linear_model
model = linear_model.LinearRegression()
model.fit (X_train,y_train)
LinearRegression()
model.score(X_test,y_test)
1.0
def predict(inputs,model):
x = np.array(inputs)
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
X_test[85]
array([49.418834 , 8.0698385 , 1.713948 , 0.0714145 , 0.63558905,
0.1142632 ])
ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145


P_huber.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 24
len(X_test)
114
import warnings
warnings.filterwarnings('ignore')
from sklearn import linear_model
model = linear_model.HuberRegressor()
model.fit (X_train,y_train)
HuberRegressor()
model.score(X_test,y_test)
1.0
def predict(inputs,model):
x = np.array(inputs)
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
X_test[85]
array([49.418834 , 8.0698385 , 1.713948 , 0.0714145 , 0.63558905,
0.1142632 ])



M_linear.ipynb
File Edit View Insert Runtime Tools Help Last edited on March 24
from sklearn import linear_model
model = linear_model.LinearRegression()
model.fit (X_train,y_train)
LinearRegression()
model.score(X_test,y_test)
1.0
def predict(inputs,model):
x = np.array(inputs)
x = np.expand_dims(x,axis=0)
return model.predict(x)[0]
X_test[85]
array([49.418834 , 8.0698385 , 1.713948 , 0.0714145 , 0.63558905,
0.1142632 ])
ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632




```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

```
len(x_train)
```

171

```
len(x_test)
```

114

```
from sklearn import linear_model
model = linear_model.Ridge()
model.fit (x_train,y_train)
```

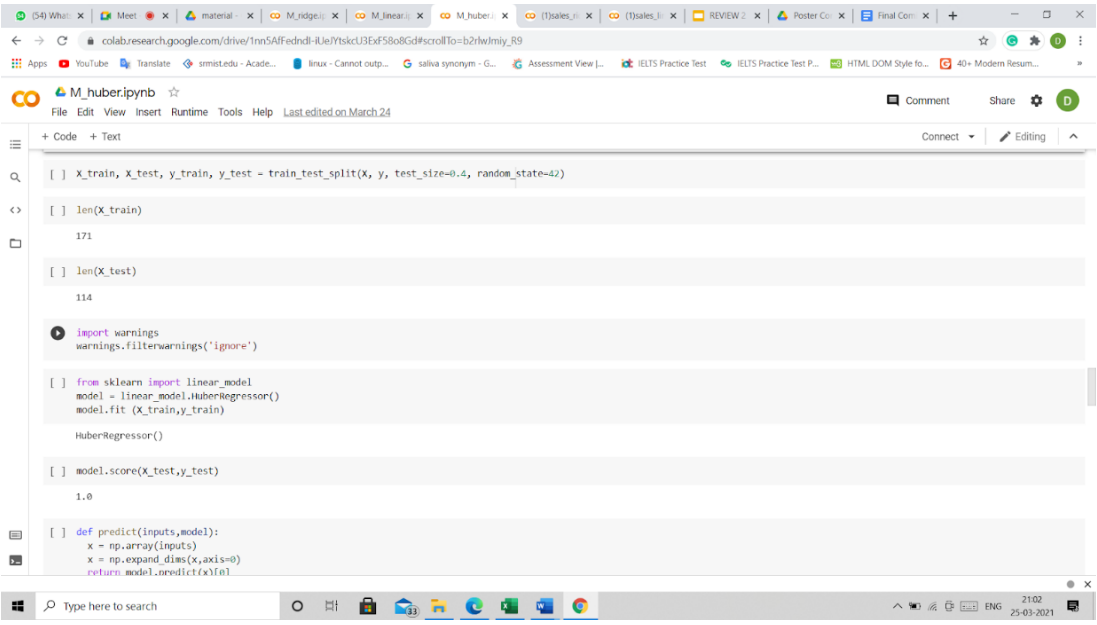
Ridge()

```
model.score(x_test,y_test)
```

1.0

```
def predict(inputs,model):
    x = np.array(inputs)
    x = np.expand_dims(x,axis=0)
    return model.predict(x)[0]
```

```
x_test[85]
```




```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)
```

```
len(x_train)
```

171

```
len(x_test)
```

114

```
import warnings
warnings.filterwarnings('ignore')
```

```
from sklearn import linear_model
model = linear_model.HuberRegressor()
model.fit (x_train,y_train)
```

HuberRegressor()

```
model.score(x_test,y_test)
```

1.0

```
def predict(inputs,model):
    x = np.array(inputs)
    x = np.expand_dims(x,axis=0)
    return model.predict(x)[0]
```

# APPENDIX 2 - MAIN CODE

Here is the main code of the project that was implemented in Google Colab.

GREEN HOUSE POLLUTION USING HUBER REGRESSION:

```
from google.colab import drive
drive.mount('/content/drive')
!pip uninstall scikit-learn
!pip -q install -U scikit-learn==0.24
path =
'/content/drive/MyDrive/Siddhant/sal
es'
import pandas as pd
df = pd.read_excel(path,
index_col=None)
df.columns
df.head()

dataset = pd.DataFrame()
dataset['Process'] = df['Process']
dataset['Transport'] =
df['Transportation']
dataset['Material'] = df['Material
Resource']
dataset

df.columns

dataset['Carbon Dioxide - Fossil '] =
df['Carbon Dioxide - Fossil ']
dataset['Methane'] = df['Methane']
dataset['Nitrous Oxide'] = df['Nitrous
Oxide ']
dataset['Methyl Bromide'] =
df['Methyl Bromide ']
dataset['Methyl Chloride'] =
df['Methyl Chloride ']
dataset['Trichloroethane'] =
df['Trichloroethane']
dataset['Chloroform'] =
df['Chloroform']
dataset['Carbon Tetrachloride'] =
df['Carbon Tetrachloride ']
dataset['CFC 13'] = df['CFC 13
(Methane, trichlorofluoro-)   ']
dataset['HCFC-22'] = df['HCFC-22
']

dataset

output_df =
dataset.drop(columns=['Process','Tra
nsport','Material'])
output_df
output_df.columns
pollutants = ['Carbon Dioxide - Fossil ',
'Methane', 'Nitrous Oxide',
       'Methyl Bromide', 'Methyl
Chloride', 'Trichloroethane',
'Chloroform',
       'Carbon Tetrachloride', 'CFC 13',
'HCFC-22']

feature_columns =
['Process','Transport','Material']
target_columns = pollutants
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()
X[0]
y1[0]
len(X)
len(y)

#REMOVING RANDOMNESS
X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]
y_train[0]
len(X_train)
len(X_test)

from sklearn.multioutput import
MultiOutputRegressor
from sklearn.linear_model import
HuberRegressor
model =
MultiOutputRegressor(HuberRegress
or()).fit(X_train, y_train)

#import sklearn
#model =
sklearn.linear_model.LinearRegressio
n().fit(X_train,y_train)
model.score(X_test,y_test)
import numpy as np
def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return pred[0]
X_test[85]
process = 53.855775
transport = 2.0971875
material = 84.390825
ftrs = [process,transport,material]
pollutant_prediction =
predict(ftrs,model)
pollutant_prediction
y_test[85]
greenhouse =
pd.DataFrame(X_test,columns=featur
e_columns)
greenhouse
preds = model.predict(X_test)
predictions =
pd.DataFrame(preds,columns=target_
columns)
predictions
l = list(predictions.columns)
for i in l:
  greenhouse[i] = predictions[i]
greenhouse
greenhouse.to_csv('greenhouse_hub
er.csv',index=False)
from google.colab import files
files.download('greenhouse_huber.cs
v')
```

GREEN HOUSE POLLUTION USING LINEAR REGRESSION:

```
dataset['Carbon Dioxide - Fossil '] =
df['Carbon Dioxide - Fossil ']
dataset['Methane'] = df['Methane']
dataset['Nitrous Oxide'] = df['Nitrous
Oxide ']
dataset['Methyl Bromide'] =
df['Methyl Bromide ']
dataset['Methyl Chloride'] =
df['Methyl Chloride ']
dataset['Trichloroethane'] =
df['Trichloroethane']
dataset['Chloroform'] =
df['Chloroform']
dataset['Carbon Tetrachloride'] =
df['Carbon Tetrachloride ']
dataset['CFC 13'] = df['CFC 13
(Methane, trichlorofluoro-)   ']
dataset['HCFC-22'] = df['HCFC-22
']

dataset

output_df =
dataset.drop(columns=['Process','Tra
nsport','Material'])


output_df

output_df.columns

pollutants = ['Carbon Dioxide - Fossil ',
'Methane', 'Nitrous Oxide',
       'Methyl Bromide', 'Methyl
Chloride', 'Trichloroethane',
'Chloroform',
       'Carbon Tetrachloride', 'CFC 13',
'HCFC-22']

feature_columns =
['Process','Transport','Material']
target_columns = pollutants
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X[0]

len(X)

len(y)
```

```
#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

len(X_train)

len(X_test)

from sklearn import linear_model
model =
linear_model.LinearRegression()
model.fit (X_train,y_train)

#import sklearn
#model =
sklearn.linear_model.LinearRegressio
n().fit(X_train,y_train)

model.score(X_test,y_test)

import numpy as np

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return pred[0]

X_test[85]

process = 53.855775
transport = 2.0971875
material = 84.390825

ftrs = [process,transport,material]

pollutant_prediction =
predict(ftrs,model)

pollutant_prediction

y_test[85]

greenhouse =
pd.DataFrame(X_test,columns=featur
e_columns)

greenhouse

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)
predictions
l = list(predictions.columns)
for i in l:
  greenhouse[i] = predictions[i]
greenhouse
greenhouse.to_csv('greenhouselinear
.csv',index=False)
from google.colab import files
files.download('greenhouselinear.csv'
)
```

---

**GREEN HOUSE POLLUTION USING RIDGE REGRESSION:**

```
dataset['Carbon Dioxide - Fossil '] =
df['Carbon Dioxide - Fossil ']
dataset['Methane'] = df['Methane']
dataset['Nitrous Oxide'] = df['Nitrous
Oxide ']
dataset['Methyl Bromide'] =
df['Methyl Bromide ']
dataset['Methyl Chloride'] =
df['Methyl Chloride ']
dataset['Trichloroethane'] =
df['Trichloroethane']
dataset['Chloroform'] =
df['Chloroform']
dataset['Carbon Tetrachloride'] =
df['Carbon Tetrachloride ']
dataset['CFC 13'] = df['CFC 13
(Methane, trichlorofluoro-)   ']
dataset['HCFC-22'] = df['HCFC-22
']

dataset

output_df =
dataset.drop(columns=['Process','Tra
nsport','Material'])


output_df

output_df.columns

pollutants = ['Carbon Dioxide - Fossil ',
'Methane', 'Nitrous Oxide',
       'Methyl Bromide', 'Methyl
Chloride', 'Trichloroethane',
'Chloroform',
       'Carbon Tetrachloride', 'CFC 13',
'HCFC-22']

feature_columns =
['Process','Transport','Material']
target_columns = pollutants
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X[0]

y1[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

len(X_train)

len(X_test)

from sklearn.multioutput import
MultiOutputRegressor
from sklearn.linear_model import
HuberRegressor
model =
MultiOutputRegressor(HuberRegress
or()).fit(X_train, y_train)

#import sklearn
#model =
sklearn.linear_model.LinearRegressio
n().fit(X_train,y_train)

model.score(X_test,y_test)

import numpy as np

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return pred[0]

X_test[85]

process = 53.855775
transport = 2.0971875
material = 84.390825

ftrs = [process,transport,material]

pollutant_prediction =
predict(ftrs,model)

pollutant_prediction

y_test[85]

greenhouse =
pd.DataFrame(X_test,columns=featur
e_columns)

greenhouse

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

l = list(predictions.columns)
for i in l:
  greenhouse[i] = predictions[i]

greenhouse

greenhouse.to_csv('greenhouse_hub
er.csv',index=False)

from google.colab import files
files.download('greenhouse_huber.cs
v')
```

---

**NONGREEN HOUSE POLLUTION USING HUBER REGRESSION:**

```
dataset['Carbon Dioxide - Non Fossil ']
= df['Carbon Dioxide - Non Fossil ']
```

```
dataset['Particulates (unspecified) '] =
df['Particulates (unspecified) ']
dataset['Particulate Matter (10) '] =
df['Particulate Matter (10) ']
dataset['Particulate Matter (2.5) '] =
df['Particulate Matter (2.5) ']
dataset['VOC (unspecified)'] = df['VOC
(unspecified)']
dataset['Carbon Monoxide'] =
df['Carbon Monoxide']
dataset['Sulfur Dioxide'] = df['Sulfur
Dioxide']
dataset['Nitrogen Oxides'] =
df['Nitrogen Oxides']

dataset

output_df =
dataset.drop(columns=['Process','Tra
nsport','Material'])


output_df

output_df.columns

pollutants = ['Carbon Dioxide - Non
Fossil ', 'Particulates (unspecified) ',
      'Particulate Matter (10) ',
'Particulate Matter (2.5) ',
      'VOC (unspecified)', 'Carbon
Monoxide', 'Sulfur Dioxide',
      'Nitrogen Oxides']

feature_columns =
['Process','Transport','Material']
target_columns = pollutants
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X[0]

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

len(X_train)

len(X_test)

from sklearn.multioutput import
MultiOutputRegressor
from sklearn.linear_model import
HuberRegressor
model =
MultiOutputRegressor(HuberRegress
or()).fit(X_train, y_train)

#import sklearn
#model =
sklearn.linear_model.LinearRegressio
n().fit(X_train,y_train)

model.score(X_test,y_test)

import numpy as np

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return pred[0]

X_test[85]

process = 53.855775
transport = 2.0971875
material = 84.390825

ftrs = [process,transport,material]

pollutant_prediction =
predict(ftrs,model)

pollutant_prediction

y_test[85]

nongreenhouse =
pd.DataFrame(X_test,columns=featur
e_columns)

nongreenhouse

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

l = list(predictions.columns)
for i in l:
  nongreenhouse[i] = predictions[i]
nongreenhouse
nongreenhouse.to_csv('nongreenhou
se_huber.csv',index=False)

from google.colab import files
files.download('nongreenhouse_hube
r.csv')
```

NONGREEN HOUSE POLLUTION USING RIDGE REGRESSION:

```
dataset['Carbon Dioxide - Non Fossil ']
= df['Carbon Dioxide - Non Fossil ']
dataset['Particulates (unspecified) '] =
df['Particulates (unspecified) ']
dataset['Particulate Matter (10) '] =
df['Particulate Matter (10) ']
dataset['Particulate Matter (2.5) '] =
df['Particulate Matter (2.5) ']
dataset['VOC (unspecified)'] = df['VOC
(unspecified)']
dataset['Carbon Monoxide'] =
df['Carbon Monoxide']
dataset['Sulfur Dioxide'] = df['Sulfur
Dioxide']
dataset['Nitrogen Oxides'] =
df['Nitrogen Oxides']

dataset

output_df =
dataset.drop(columns=['Process','Tra
nsport','Material'])


output_df

output_df.columns

pollutants = ['Carbon Dioxide - Non
Fossil ', 'Particulates (unspecified) ',
      'Particulate Matter (10) ',
'Particulate Matter (2.5) ',
      'VOC (unspecified)', 'Carbon
Monoxide', 'Sulfur Dioxide',
      'Nitrogen Oxides']

feature_columns =
['Process','Transport','Material']
target_columns = pollutants
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X[0]

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.Ridge()
model.fit (X_train,y_train)

#import sklearn
#model =
sklearn.linear_model.LinearRegressio
n().fit(X_train,y_train)

model.score(X_test,y_test)

import numpy as np
```

```
def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return pred[0]

X_test[85]

process = 53.855775
transport = 2.0971875
material = 84.390825

ftrs = [process,transport,material]

pollutant_prediction =
predict(ftrs,model)

pollutant_prediction

y_test[85]

nongreenhouse =
pd.DataFrame(X_test,columns=featur
e_columns)

nongreenhouse

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

l = list(predictions.columns)
for i in l:
  nongreenhouse[i] = predictions[i]
nongreenhouse
nongreenhouse.to_csv('nongreenhou
se_ridge.csv',index=False)

from google.colab import files
files.download('nongreenhouse_ridge
.csv')
```

NONGREEN HOUSE POLLUTION USING LINEAR REGRESSION:

```
dataset['Carbon Dioxide - Non Fossil ']
= df['Carbon Dioxide - Non Fossil ']
dataset['Particulates (unspecified) '] =
df['Particulates (unspecified) ']
dataset['Particulate Matter (10) '] =
df['Particulate Matter (10) ']
dataset['Particulate Matter (2.5) '] =
df['Particulate Matter (2.5) ']
dataset['VOC (unspecified)'] = df['VOC
(unspecified)']
dataset['Carbon Monoxide'] =
df['Carbon Monoxide']
dataset['Sulfur Dioxide'] = df['Sulfur
Dioxide']
dataset['Nitrogen Oxides'] =
df['Nitrogen Oxides']

dataset

output_df =
dataset.drop(columns=['Process','Tra
nsport','Material'])

output_df

output_df.columns

pollutants = ['Carbon Dioxide - Non
Fossil ', 'Particulates (unspecified) ',
      'Particulate Matter (10) ',
'Particulate Matter (2.5) ',
      'VOC (unspecified)', 'Carbon
Monoxide', 'Sulfur Dioxide',
      'Nitrogen Oxides']

feature_columns =
['Process','Transport','Material']
target_columns = pollutants
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X[0]

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

len(X_train)

len(X_test)

from sklearn import linear_model
model =
linear_model.LinearRegression()
model.fit (X_train,y_train)

#import sklearn
#model =
sklearn.linear_model.LinearRegressio
n().fit(X_train,y_train)

model.score(X_test,y_test)

import numpy as np

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return pred[0]
X_test[85]

process = 53.855775
transport = 2.0971875
material = 84.390825
ftrs = [process,transport,material]
pollutant_prediction =
predict(ftrs,model)
pollutant_prediction
y_test[85]
nongreenhouse =
pd.DataFrame(X_test,columns=featur
e_columns)
nongreenhouse
preds = model.predict(X_test)
predictions =
pd.DataFrame(preds,columns=target_
columns)
predictions
l = list(predictions.columns)
for i in l:
  nongreenhouse[i] = predictions[i]
nongreenhouse
nongreenhouse.to_csv('nongreenhou
se_linear.csv',index=False)
from google.colab import files
files.download('nongreenhouse_linea
r.csv')
```

PTM HUBER REGRESSION:

```
from google.colab import drive
drive.mount('/content/drive')

!pip uninstall scikit-learn
!pip -q install -U scikit-learn==0.24

path =
'/content/drive/MyDrive/Siddhant/sal
es'
import pandas as pd
df = pd.read_excel(path,
index_col=None)

df.columns

df.head()

dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Process'] = df['Process']
dataset['Transportation'] =
df['Transportation']
dataset['Material Resource'] =
df['Material Resource']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns =
['Process','Transportation','Material
Resource']
```

```
X =
dataset[feature_columns].to_numpy(
)
y1 = dataset['Process'].to_numpy()
y2 =
dataset['Transportation'].to_numpy()
y3 = dataset['Material
Resource'].to_numpy()
y =
dataset[target_columns].to_numpy()

X

y1[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y1_train[0]

len(X_train)

len(X_test)

from sklearn.multioutput import
MultiOutputRegressor
from sklearn.linear_model import
HuberRegressor
model =
MultiOutputRegressor(HuberRegress
or()).fit(X_train, y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return
pred[0][0],pred[0][1],pred[0][2]

X_test[85]

ngas = 116.1003
petroleum = 18.958575
coal = 4.0266
hydro =  0.167775
nuclear = 1.4931975
others = 0.26844

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

process,transportation,material_reso
urce = predict(ftrs,model)
print(process,transportation,material
_resource)
y_test[85]

ptm_huber =
pd.DataFrame(X_test,columns=featur
e_columns)
ptm_huber
preds = model.predict(X_test)
predictions =
pd.DataFrame(preds,columns=target_
columns)
ptm_huber['Process'] =
predictions['Process']
ptm_huber['Transportation'] =
predictions['Transportation']
ptm_huber['Material Resource'] =
predictions['Material Resource']

ptm_huber

ptm_huber.to_csv('ptm_huber.csv',in
dex=False)
from google.colab import files
files.download('ptm_huber.csv')
```

PTM LINEAR REGRESSION:

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Process'] = df['Process']
dataset['Transportation'] =
df['Transportation']
dataset['Material Resource'] =
df['Material Resource']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns =
['Process','Transportation','Material
Resource']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model =
linear_model.LinearRegression()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return
pred[0][0],pred[0][1],pred[0][2]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

process,transportation,material_reso
urce = predict(ftrs,model)

print(process,transportation,material
_resource)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Process'] =
predictions['Process']
energy['Transportation'] =
predictions['Transportation']
energy['Material Resource'] =
predictions['Material Resource']
```

```
energy

energy.to_csv('energy_test.csv',index
=False)

from google.colab import files
files.download('energy_test.csv')
```

PTM RIDGE REGRESSION:

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Process'] = df['Process']
dataset['Transportation'] =
df['Transportation']
dataset['Material Resource'] =
df['Material Resource']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns =
['Process','Transportation','Material
Resource']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.Ridge()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  pred = model.predict(x)
  return
pred[0][0],pred[0][1],pred[0][2]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

process,transportation,material_reso
urce = predict(ftrs,model)

print(process,transportation,material
_resource)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Process'] =
predictions['Process']
energy['Transportation'] =
predictions['Transportation']
energy['Material Resource'] =
predictions['Material Resource']

energy

energy.to_csv('ptm_ridge.csv',index=
False)

from google.colab import files
files.download('ptm_ridge.csv')
```

P HUBER REGRESSION:

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Process'] = df['Process']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Process']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

import warnings
warnings.filterwarnings('ignore')

from sklearn import linear_model
model =
linear_model.HuberRegressor()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]
```

```
ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

process = predict(ftrs,model)

print(process)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Process'] =
predictions['Process']

energy

energy.to_csv('p_huber.csv',index=Fal
se)

from google.colab import files
files.download('p_huber.csv')
```

P LINEAR REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Process'] = df['Process']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Process']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model =
linear_model.LinearRegression()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

process = predict(ftrs,model)

print(process)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Process'] =
predictions['Process']

energy

energy.to_csv('p_linear.csv',index=Fal
se)

from google.colab import files
files.download('p_linear.csv')
```

P RIDGE REGRESSION:

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Process'] = df['Process']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Process']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)
```

```
len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.Ridge()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

process = predict(ftrs,model)

print(process)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Process'] =
predictions['Process']

energy

energy.to_csv('p_ridge.csv',index=Fals
e)

from google.colab import files
files.download('p_ridge.csv')
```

T HUBER REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Transportation'] =
df['Transportation']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Transportation']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

import warnings
warnings.filterwarnings('ignore')

from sklearn import linear_model
model =
linear_model.HuberRegressor()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

transportation = predict(ftrs,model)

print(transportation)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Transportation'] =
predictions['Transportation']

energy

energy.to_csv('T_huber.csv',index=Fal
se)

from google.colab import files
files.download('T_huber.csv')
```

T LINEAR REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Transportation'] =
df['Transportation']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Transportation']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]
```

```
len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model =
linear_model.LinearRegression()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

transportation = predict(ftrs,model)

print(transportation)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Transportation'] =
predictions['Transportation']

energy

energy.to_csv('t_linear.csv',index=Fal
se)

from google.colab import files
files.download('t_linear.csv')
```

T RIDGE REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Transportation'] =
df['Transportation']

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Transportation']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.Ridge()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

transportation = predict(ftrs,model)

print(transportation)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Transportation'] =
predictions['Transportation']

energy

energy.to_csv('t_ridge.csv',index=Fals
e)

from google.colab import files
files.download('t_ridge.csv')
```

M HUBER REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset

dataset['Material Resource'] =
df['Material Resource']

dataset
```

```
import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Material
Resource']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

import warnings
warnings.filterwarnings('ignore')

from sklearn import linear_model
model =
linear_model.HuberRegressor()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

material = predict(ftrs,model)

print(material)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions

energy['Material Resource'] =
predictions['Material Resource']

energy

energy.to_csv('m_huber.csv',index=F
alse)

from google.colab import files
files.download('m_huber.csv')
```

---

M LINEAR REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural
Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset['Material Resource'] =
df['Material Resource']

dataset

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','
Nuclear','Others']
target_columns = ['Material
Resource']
X =
dataset[feature_columns].to_numpy(
)
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model =
linear_model.LinearRegression()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,o
thers]

Material = predict(ftrs,model)

print(Material)

y_test[85]

energy =
pd.DataFrame(X_test,columns=featur
e_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_
columns)

predictions
```

```
energy['Material Resource'] =
predictions['Material Resource']

energy

energy.to_csv('m_ridge.csv',index=False)

from google.colab import files
files.download('m_ridge.csv')
```

M RIDGE REGRESSION

```
dataset = pd.DataFrame()
dataset['Natural Gas'] = df['Natural Gas ']
dataset['Petroleum'] = df['Petroleum']
dataset['Coal'] = df['Coal']
dataset['Hydropower'] =
df['Hydropower']
dataset['Nuclear'] = df['Nuclear']
dataset['Others'] = df['Others']

dataset['Material Resource'] =
df['Material Resource']

dataset

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

feature_columns = ['Natural
Gas','Petroleum','Coal','Hydropower','Nuclear','Others']
target_columns = ['Material
Resource']
X =
dataset[feature_columns].to_numpy()
y =
dataset[target_columns].to_numpy()

X

y[0]

len(X)

len(y)

#REMOVING RANDOMNESS

X_train = X[:171,]
X_test = X[171:,]
y_train = y[:171,]
y_test = y[171:,]

X_train[0]

y_train[0]

#WITH RANDOMNESS

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.Ridge()
model.fit (X_train,y_train)

model.score(X_test,y_test)

def predict(inputs,model):
  x = np.array(inputs)
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[85]

ngas = 49.418834
petroleum = 8.0698385
coal = 1.713948
hydro = 0.0714145
nuclear = 0.63558905
others = 0.1142632

ftrs =
[ngas,petroleum,coal,hydro,nuclear,others]

Material = predict(ftrs,model)

print(Material)

y_test[85]

energy =
pd.DataFrame(X_test,columns=feature_columns)

energy

preds = model.predict(X_test)

predictions =
pd.DataFrame(preds,columns=target_columns)

predictions

energy['Material Resource'] =
predictions['Material Resource']

energy

energy.to_csv('m_ridge.csv',index=False)

from google.colab import files
files.download('m_ridge.csv')
```

SALES WITHOUT VAT RIDGE

```
features =
pd.DataFrame(columns=['plastics','net_proceeds'])
features['plastics'] = df['Number of
single use plastic bags issued']
features['net_proceeds'] = df['Net
proceeds  (£) (Gross less VAT)']

features

features['target'] = df['Gross proceeds
of charge (£)']
features

dataset = features.copy()

dataset

import numpy as np
from sklearn.model_selection import
train_test_split

X =
dataset[['plastics','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()

X

y

len(X)

len(y)

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.4,
random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model= linear_model.Ridge()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(input1,input2,model):
  x = np.array([input1,input2])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[75]

plastics = 237974
net_proceeds = 9916

gross_proceeds =
predict(plastics,net_proceeds,model)

gross_proceeds

y_test[75]

preds = model.predict(X_test)

preds[75]

sales =
pd.DataFrame(X_test,columns=['Number of single use plastic bags
issued','Net proceeds'])

k =[]
for i in preds:
  k.append(
float("{:.4f}".format(float(i))))

sales['Gross Proceeds'] = k

sales
```

```
sales.to_csv('saleswithoutvat_test_ridge.csv',index=False)

from google.colab import files
files.download('saleswithoutvat_test_ridge.csv')
```

SALES WITHOUT VAT LINEAR

```
features = pd.DataFrame(columns=['plastics','net_proceeds'])
features['plastics'] = df['Number of single use plastic bags issued']
features['net_proceeds'] = df['Net proceeds  (£) (Gross less VAT)']

features

features['target'] = df['Gross proceeds of charge (£)']
features

dataset = features.copy()


dataset

import numpy as np
from sklearn.model_selection import train_test_split

X = dataset[['plastics','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()

X

y

len(X)

len(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model= linear_model.LinearRegression()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(input1,input2,model):
  x = np.array([input1,input2])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[75]

plastics = 237974
net_proceeds = 9916

gross_proceeds = predict(plastics,net_proceeds,model)

gross_proceeds

y_test[75]

preds = model.predict(X_test)

preds[75]

sales = pd.DataFrame(X_test,columns=['Number of single use plastic bags issued','Net proceeds'])

k =[]
for i in preds:
  k.append( float("{:.4f}".format(float(i))))

sales['Gross Proceeds'] = k

sales

sales.to_csv('saleswithoutvat_test_linear.csv',index=False)

from google.colab import files
files.download('saleswithoutvat_test_linear.csv')
```

SALES WITHOUT VAT HUBER

```
features = pd.DataFrame(columns=['plastics','net_proceeds'])
features['plastics'] = df['Number of single use plastic bags issued']
features['net_proceeds'] = df['Net proceeds  (£) (Gross less VAT)']

features

features['target'] = df['Gross proceeds of charge (£)']
features

dataset = features.copy()


dataset

import numpy as np
from sklearn.model_selection import train_test_split

X = dataset[['plastics','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()

X

y

len(X)

len(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.HuberRegressor()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(input1,input2,model):
  x = np.array([input1,input2])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[75]

plastics = 237974
net_proceeds = 9916

gross_proceeds = predict(plastics,net_proceeds,model)

gross_proceeds

y_test[75]

preds = model.predict(X_test)

preds[75]

sales = pd.DataFrame(X_test,columns=['Number of single use plastic bags issued','Net proceeds'])

k =[]
for i in preds:
  k.append( float("{:.4f}".format(float(i))))

sales['Gross Proceeds'] = k

sales

sales.to_csv('saleswithoutvat_test_huber.csv',index=False)

from google.colab import files
files.download('saleswithoutvat_test_huber.csv')
```

SALES WITH VAT LINEAR

```
features = pd.DataFrame(columns=['plastics','VAT','net_proceeds'])
features['plastics'] = df['Number of single use plastic bags issued']
features['VAT'] = df['VAT (£)']
features['net_proceeds'] = df['Net proceeds  (£) (Gross less VAT)']

features

features['target'] = df['Gross proceeds of charge (£)']
features

dataset = features.copy()

dataset

import numpy as np
```

```
from sklearn.model_selection import train_test_split

X = dataset[['plastics','VAT','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()

X

y

len(X)

len(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.LinearRegression()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(input1,input2,input3,model):
  x = np.array([input1,input2,input3])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[75]

plastics = 237974
VAT = 1983
net_proceeds = 9916

gross_proceeds = predict(plastics,VAT,net_proceeds,model)  /predict one row row for demo

gross_proceeds

y_test[75]

preds = model.predict(X_test)

preds[75]  //for prediction of whole table

sales = pd.DataFrame(X_test,columns=['Number of single use plastic bags issued','VAT','Net proceeds'])

k =[]
for i in preds:
  k.append( float("{:.4f}".format(float(i))))

sales['Gross Proceeds'] = k

sales

sales.to_csv('sales_test_linear.csv',index=False)

from google.colab import files
files.download('sales_test_linear.csv')
```

SALES WITH VAT HUBER

```
features = pd.DataFrame(columns=['plastics','VAT','net_proceeds'])
features['plastics'] = df['Number of single use plastic bags issued']
features['VAT'] = df['VAT (£)']
features['net_proceeds'] = df['Net proceeds  (£) (Gross less VAT)']

features

features['target'] = df['Gross proceeds of charge (£)']
features

dataset = features.copy()


dataset

import numpy as np
from sklearn.model_selection import train_test_split

X = dataset[['plastics','VAT','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()

X

y

len(X)

len(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=42)

len(X_train)

len(X_test)

from sklearn import linear_model
model = linear_model.HuberRegressor()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def predict(input1,input2,input3,model):
  x = np.array([input1,input2,input3])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[75]

plastics = 237974
VAT = 1983
net_proceeds = 9916

gross_proceeds = predict(plastics,VAT,net_proceeds,model)

gross_proceeds

y_test[75]

preds = model.predict(X_test)

preds[75]

sales = pd.DataFrame(X_test,columns=['Number of single use plastic bags issued','VAT','Net proceeds'])

k =[]
for i in preds:
  k.append( float("{:.4f}".format(float(i))))

sales['Gross Proceeds'] = k

sales

sales.to_csv('sales_test_huber.csv',index=False)

from google.colab import files
files.download('sales_test_huber.csv')
```

SALES WITH VAT RIDGE

```
features = pd.DataFrame(columns=['plastics','VAT','net_proceeds'])
features['plastics'] = df['Number of single use plastic bags issued']
features['VAT'] = df['VAT (£)']
features['net_proceeds'] = df['Net proceeds  (£) (Gross less VAT)']

features

features['target'] = df['Gross proceeds of charge (£)']
features

dataset = features.copy()

dataset

import numpy as np
from sklearn.model_selection import train_test_split

X = dataset[['plastics','VAT','net_proceeds']].to_numpy()
y = dataset['target'].to_numpy()

X

y

len(X)

len(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.7, random_state=42)
```

```
from sklearn import linear_model
model = linear_model.Ridge()
model.fit(X_train,y_train)

model.score(X_test,y_test)

def
predict(input1,input2,input3,model):
  x = np.array([input1,input2,input3])
  x = np.expand_dims(x,axis=0)
  return model.predict(x)[0]

X_test[75]

plastics = 237974
VAT = 1983
net_proceeds = 9916

gross_proceeds =
predict(plastics,VAT,net_proceeds,mo
del)

gross_proceeds

y_test[75]

preds = model.predict(X_test)

preds[1]

y_test[1]

sales =
pd.DataFrame(X_test,columns=['Num
ber of single use plastic bags
issued','VAT','Net proceeds'])

k =[]
for i in preds:
  k.append(
float("{:.4f}".format(float(i))))

sales['Gross Proceeds'] = k

sales

sales.to_csv('sales_test_ridge.csv',ind
ex=False)

from google.colab import files
files.download('sales_test_ridge.csv')
```

# PLAGARISM REPORT

**PUBLICATION PROOF**

**Published.** Suparno Bhatta, Debanjana Chakraborty and Poovammal E, Arriving at Optimal Cost for a Sustainable Product using an Integrated LCA-SCM Model, Journal of Information Technology in Industry, Vol. 9, No. 2, 2021, pp 1193-1199, DOI: https://doi.org/10.17762/itii.v9i2.472

**Journal Website:** http://www.it-in-industry.org/index.php/itii

**Indexing:** Web of Science, ISSN, Crossref, publons.

**Tech Blog Published.** to 'Towards Data Science'

Link: **https://towardsdatascience.com/application-of-an-integrated-life-cycle-assessment-and-supply-chain-analytics-on-ldpe-6aba41555921**

**Manuscript Submitted.** to Journal of Management Information Systems.

**Indexing:** EBSCOhost, Compendex, Scopus, Ovid, ProQuest, SCIMP, Web of Science, etc.

**Journal Website:** **https://www.jmis-web.org/** (Print ISSN: 0742-1222 Online ISSN: 1557-928X)

# PLAGARISM PROOF

report-24th may

ORIGINALITY REPORT

| 6% | 3% | 3% | 2% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

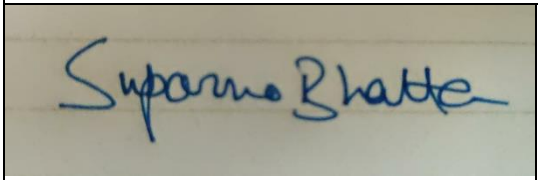
| | | |
|---|---|---|
| 1 | it-in-industry.org<br>Internet Source | 1% |
| 2 | Submitted to University of Leeds<br>Student Paper | <1% |
| 3 | E. Cobas, C. Hendrickson, L. Lave, F. McMichael. "Economic input/output analysis to aid life cycle assessment of electronics products", Proceedings of the 1995 IEEE International Symposium on Electronics and the Environment ISEE (Cat. No.95CH35718), 1995<br>Publication | <1% |
| 4 | Subramanian Senthilkannan Muthu. "Calculating the water and energy footprints of textile products", Elsevier BV, 2020<br>Publication | <1% |
| 5 | patents.google.com<br>Internet Source | <1% |
| 6 | en.wikipedia.org<br>Internet Source | <1% |
| 7 | Zainab-L Idris, Ong Su Yean, Sudesh Kumar. "Polyhydroxyalkanoates", Wiley, 2018<br>Publication | <1% |
| 8 | Jung-Hyun Lee, Maher Ibrahim Sameen, Biswajeet Pradhan, Hyuck-Jin Park. "Modeling landslide susceptibility in data-scarce environments using optimized data mining and statistical methods", Geomorphology, 2018<br>Publication | <1% |
| 9 | Carlo Brondi, Federico Fragassi, Tommaso Pasetti, Rosanna Fornasiero. "Evaluating sustainability trade-offs along supply chain", 2014 International Conference on Engineering, Technology and Innovation (ICE), 2014<br>Publication | <1% |
| 10 | Longwen Chen, Evelyne Gonze, Michel Ondarts, Jonathan Outin, Yves Gonthier. "Electrostatic precipitator for fine and ultrafine particle removal from indoor air environments", Separation and Purification Technology, 2020<br>Publication | <1% |
| 11 | Submitted to Universitat Politècnica de València<br>Student Paper | <1% |
| 12 | Submitted to University of Dallas<br>Student Paper | <1% |
| 13 | www.trigonplus.cz<br>Internet Source | <1% |
| 14 | C. Brondi, E. Carpanzano. "A modular framework for the LCA-based simulation of production systems", CIRP Journal of Manufacturing Science and Technology, 2011<br>Publication | <1% |
| 15 | Submitted to Whitgift School<br>Student Paper | <1% |
| 16 | Submitted to Swinburne University of Technology<br>Student Paper | <1% |
| 17 | Submitted to University of Pretoria<br>Student Paper | <1% |
| 18 | www.americanchemistry.com<br>Internet Source | <1% |
| 19 | www.conserve-energy-future.com<br>Internet Source | <1% |
| 20 | Zhiqiang Liu, Danni Gong, Linlin Wang. "Heat transfer with kinetic phase transition in geomaterials", International Journal of Heat and Mass Transfer, 2021<br>Publication | <1% |
| 21 | oro.open.ac.uk<br>Internet Source | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | On | Exclude matches | < 8 words |
| Exclude bibliography | On | | |

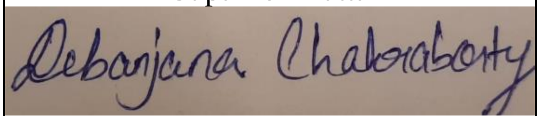
<table>
<tr><td colspan="5">SRM INSTITUTE OF SCIENCE AND TECHNOLOGY<br>(Deemed to be University u/s 3 of UGC Act, 1956)</td></tr>
<tr><td colspan="5">Office of Controller of Examinations</td></tr>
<tr><td colspan="5">REPORT FOR PLAGIARISM CHECK ON THE DISSERTATION / PROJECT REPORTS FOR UG/ PG PROGRAMMES</td></tr>
<tr><td>1</td><td>Name of the Candidate</td><td colspan="3">Suparno Bhatta</td></tr>
<tr><td>2</td><td>Address of the Candidate</td><td colspan="3">Email: sp7743@srmist.edu.in<br>Mobile Number: 9994088010</td></tr>
<tr><td>3</td><td>Registration Number</td><td colspan="3">RA1711003010172</td></tr>
<tr><td>4</td><td>Date of Birth</td><td colspan="3">06/07/1999</td></tr>
<tr><td>5</td><td>Department</td><td colspan="3">Computer Science and Engineering</td></tr>
<tr><td>6</td><td>Faculty</td><td colspan="3">Engineering and Technology</td></tr>
<tr><td>7</td><td>Title of the Dissertation / Project</td><td colspan="3">Implementation Of Environmental And Supply Chain Analytics With Regression On Single Use Plastics</td></tr>
<tr><td>8</td><td>Whether the above project / dissertation is done by</td><td colspan="3">GROUP<br>a) If group, number of students: 2<br>b) Name and Register Numbers of other candidates:<br>Name: Debanjana Chakraborty (RA1711003010212)<br>Email: dr6016@srmist.edu.in<br>Phone: 9830471957</td></tr>
<tr><td>9</td><td>Name and address of the Supervisor / Guide</td><td colspan="3">Mail ID: poovamme@srmist.edu.in<br>Mobile Number:9444460822</td></tr>
<tr><td>10</td><td>Name and address of the Co-Supervisor / Co- Guide (if any)</td><td colspan="3">N/A</td></tr>
<tr><td>11</td><td>Software Used</td><td colspan="3">Google Colab - Python, MS Excel, Google Sheets</td></tr>
<tr><td>12</td><td>Date of Verification</td><td colspan="3">25/05/2021</td></tr>
<tr><td>13</td><td>Plagiarism Details</td><td colspan="3">{To attach the report from the software}</td></tr>
<tr><td>Chapter</td><td>Title of the</td><td>Percentage of similarity index (including self-</td><td>Percentage of similarity index</td><td>Percentage of plagiarism excluding</td></tr>
</table>

| | **Chapter** | **citations)** | **(excluding self-citations)** | **Quotes, Bibliography.** |
|---|---|---|---|---|
| 1 | INTRODUCTION | < 2 % | < 1 % | < 1 % |
| 2 | LITERATURE REVIEW | 0 % | 0 % | 0 % |
| 3 | INTEGRATED LCA-SCM MODEL | < 1 % | < 1% | < 1% |
| 4 | UML DIAGRAM | 0 % | 0 % | 0% |
| 5 | ARCHITECTURE DIAGRAM | 0 % | 0 % | 0 % |
| 6 | MODULE DESCRIPTION | 0 % | 0 % | 0 % |
| 7 | DATABASE COLLECTION | < 1 % | < 1 % | < 1 % |
| 8 | IMPLEMENTATION | < 1 % | < 1 % | < 1% |
| 9 | RESULTS AND DISCUSSIONS | < 1% | < 1 % | < 1 % |
| 10 | CONCLUSION | 0 % | 0 % | 0 % |
| Appendices | | 0 % | 0 % | 0 % |

We declare that the above information has been verified and found true to the best of our knowledge.



Suparno Bhatta



Debanjana Chakraborty

Signature of the Candidate

Name and Signature of the Staff who uses the plagiarism

Dr Poovammal E
Name and Signature of Guide:

Name and Signature of Co - Guide:

Dr. B. Amutha
Name and signature of the Head of Department